# TSPS No. 1:

# Operational Programs

By A. W. KETTLEY, E. J. PASTERNAK and M. F. SIKORSKY

(Manuscript received August 31, 1970)

*The Traffic Service Position System (TSPS) No. 1 Operational Programs provide the call processing logic which controls the handling of calls served by the system. The organization of these programs is described and the major events encountered in processing a sample call are covered in this article.*

## I. INTRODUCTION

The call processing programs of the Traffic Service Position System provide the logic and control for processing telephone calls. These programs supervise the calls, transmit and receive signals to and from other switching systems, send information to and respond to signals from operators, and record billing details on the calls. The programs that provide these and other functions are described in this article with emphasis on those areas that are new or substantially different from No. 1 ESS. To illustrate how the various programs interact and relate to hardware actions, a coin toll call is traced from origination to completion. At the end of the article some ancillary features are also described because of their novelty or because similar programs have not been previously described in the B.S.T.J.

Although the TSPS call processing programs are patterned after those of No. 1 ESS, the nature of the TSPS call dictates a number of changes in program and memory organization. Since TSPS handles toll traffic that enters the system via high-occupancy trunks, a software register is dedicated to each of the trunks to store the billing details on the calls as well as miscellaneous information required in handling the call. The high occupancy of these registers, which tends to minimize the memory-saving advantages of the traffic engineered ESS call registers, plus the simplicity of the control strategy for dedicated registers were important factors in the decision to use this

type of call register arrangement. Similar arguments apply to the position register, which is also dedicated. For the sake of uniformity all service circuit registers are similarly dedicated.

However, two other registers—the auxiliary trunk register and the path memory annex—are not dedicated. The auxiliary register is used to store details on coin calls, and the path memory annex stores information concerning a connection through the network. Since coin call details are not required on noncoin calls, and since TSPS network connections are made for a small part of the duration of a toll connection, both of these registers are engineered items and are linked to the trunk register as required.

Another difference between No. 1 ESS and TSPS is the significant amount of data exchanged between the system and the operators. This exchange is required, for instance, to set the stage for the operator on each call by identifying the type of call she is to handle and giving her other pertinent information about the call. The system in turn receives information from her as she operates keys and responds with some action to indicate the successful reception of her key signal. Since the operator has freedom to operate her keys in a variety of sequences for the same type of call, the programs that react to individual key operations are required to determine much about the nature of the call and its current state before selecting the appropriate course of action. It is also necessary to recognize invalid key sequences and to ignore them or to defer their recognition until some other action has taken place. These requirements considerably complicate the organization of the programs that respond to the operators key actions. In spite of widespread use of subroutines in this area, over 10,000 words of program are required for these functions alone.

One last major difference between No. 1 ESS and TSPS lies in the use of the network. Unlike the ESS network, the TSPS network does not switch calls but is used to attach the appropriate special-purpose circuit to the connection as needed. These circuits typically detect and identify the called and calling numbers, outpulse the called digits, collect and return coins, apply audible ringing, reorder, or a recorded announcement, and connect an operator as needed. The TSPS network requirements result in a simpler network than in ESS and a simplification in network control program strategy.

The programs associated with call processing may be classified in three categories: (*i*) input-output programs which detect and report changes in circuits outside of the processor and those which control

signals to similar external hardware units; (*ii*) call control programs which have only call-related functions and whose purpose is to advance a call to completion; and (*iii*) programs of a general nature which perform frequently used functions and which may be considered to be service routines used by call processing programs and others.

## II. INPUT-OUTPUT PROGRAMS

### 2.1 *Input Programs*

The programs which are responsible for administering inputs to the system are designed to be relatively simple, highly efficient programs which run during H- or J-level interrupts and are responsible only for the acquisition of data. The information is passed on to other base level call programs responsible for the analysis of the input data. This method of operation is used because of the large number of input sources which must be scanned periodically and from which, on the average, a relatively small amount of data is obtained.

The trunk supervisory scan program is responsible for detecting changes of state in the supervisory ferrods of circuits on universal trunk frames. Because of the variety of trunk circuits found on a universal trunk frame such as incoming trunks, delay call trunks, and operator service trunks, a change of state in a ferrod can mean several things, depending on the trunk type with which it is associated. While it is not the responsibility of the supervisory scan program to determine the meaning of a change in state, it does separate the changes into two categories for purposes of reporting to other call programs. The changes from the on-hook to off-hook state are reported directly through the trunk seizure and answer hopper. The changes from the off-hook to on-hook state represent the beginning of a potential disconnect or flash. Consequently, the trunk supervisory scan program reports the latter change of state to a hit timing program which will time the duration of the on-hook signal to discriminate between flashes and true disconnects. The results of this timing are then reported to a call control program. It is the function of the call control programs receiving these reports to interpret the data and initiate the appropriate action.

The digit scan program scans the digit receivers and transmits the digits, as received, to the appropriate call control program. The call programs analyze the digits and determine when digit reception is complete and when to terminate scanning of a particular digit receiver.

The group gate scan program periodically scans ferrods associated

with the operator groups to obtain signals from keys depressed by operators at the traffic service positions. This data is passed to call control programs which interpret the key signal and take appropriate action.

The sender attached scan program scans the ferrods in the digit transmitters which indicate when a digit receiver has been attached at the toll office and is ready to receive digits. It reports this to a call control program which then activates outpulsing of the called number to the toll office.

#### 2.1.1 *Output Programs*

When signals are to be sent from the processor to peripheral units, the system must buffer data to allow for the difference in speed at which the processor can process data and the speed at which the external circuits can operate. This buffering is accomplished by the output programs. The call control programs, which determine the control signals to be sent, store the data for these signals in various types of buffers which are then administered at the proper frequency by output programs.

In sending signals to control circuits such as trunks, link networks, etc., the call control programs load a peripheral order buffer (POB) with orders which are then distributed to the appropriate peripheral unit by the POB execution program. This program also reports to the call control program at a later time that the order was or was not successfully executed.

The call control program responsible for outpulsing digits to the toll office loads a register associated with the outpulser with a sequence of central pulse distributor addresses containing the called number. The output program associated with multifrequency outpulsing then distributes these at the rate of one digit every 133 milliseconds or 7.5 digits per second to the toll office.

The relays of the position circuits are controlled in a manner similar to those of trunks. The call control program loads a position information buffer (PIB) which is then executed by the PIB execution program.

### 2.2 *Call Control Programs*

Similar to No. 1 ESS, the programs which administer the processing of a call are separated into functions each of which is related to a stage in the progress of a call. This separation allows the programs to be of a manageable size and perform a well-defined function.

On a normal call, the programs responsible for handling the call are: (*i*) the call connections program, used to set up connections to the digit receivers, positions, and outpulsers; (*ii*) the digit analysis programs, used to record and analyze the digits received and determine the general type of call; (*iii*) the operator actions programs which are responsible for administering a call during the time it is connected to a position; and (*iv*) the disconnect program which controls the disconnect of the call, causes recording of call billing data, and returns the trunk to an idle state.

### 2.2.1 *Call Connections Program*

The call connections program is the first in the line of call control programs to handle a call. It receives the report from the supervisory trunk scan program that a trunk has been seized by the local office. To determine what must be done to serve this request, the program consults the parameter register associated with the incoming trunk.

Having obtained the necessary information, the program selects a network path and completes a connection over that path from the trunk to a digit receiver. In the case of multifrequency trunks, it also causes a signal to be sent to the local office indicating that a receiver is connected and transmission of digits can start. It finally activates digit scanning for this receiver. After digit scanning is initiated, the digit analysis program assumes control of the call.

At the completion of digit analysis, control is returned to the call connections program which determines whether automatic number identification (ANI) information is to be received from this office. If it is, the call connections program establishes a connection to a multifrequency receiver if it is not already connected and generates a signal to the local office requesting ANI digit transmission. The ANI digit analysis program now assumes control until reception of the calling party's number is completed.

When the calling party identification has been received, control is returned again to the call connections program. At this time, a general analysis is performed on the information obtained. The customer dialed digits are checked and the call is marked as 0+, 1+, etc.

There are three basic call states at this time: position assisted, position assisted customer-dialed, and customer direct distance dialing (DDD). For the DDD call, a connection is established to an outpulsing circuit and the called number will be outpulsed forward. On completion of outpulsing, the call connections program establishes the talking connection through the trunk and administers supervisory

reports so as to establish answer. Upon receiving an answer report, the call start time is loaded in the trunk register and the register is placed in the appropriate talking state so that all further reports and actions on this trunk will be handled by the proper disconnect programs.

### 2.2.2 *Digit Analysis Programs*

The digit analysis programs are composed of three routines. Two are responsible for reception and analysis of the called party's number for multifrequency and dial pulse digit trunks. The third is for reception and analysis of the ANI digits transmitted hy the local office.

2.2.2.1 *DP Digit Analysis.* The dial pulse digit analysis program counts and stores each digit as it is received in the register associated with the incoming trunk. For most digits no analysis is required, but in some situations an analysis is made of the digits that have been received. For example, when the first digit is received, it is checked to see if it is a zero or a one. Depending on the type of trunk, this could terminate dialing as a dialing error, be a valid condition, or institute timing to determine if more digits are to he received. Any other first digit is merely stored and the digit counter incremented by one. Upon receipt of the third digit, analysis of the first three digits is made. If the number 800 is received, it is known that the customer is dialing an INWATS call and that seven more digits are expected. If the first three digits are an office code in the area of the originating party, it is known that four more digits are expected. If the first three digits are a foreign area code, seven more digits are expected.

In the ahove instances, the number of digits to be expected is immediately evident and no more action is taken until the total number of digits expected are received. However, there are certain instances where a conflict must be resolved. For example, the first three digits dialed may he both an office code in the area serving the originating subscriber and a foreign area code. In this case, timing for 2.5 to 4.5 seconds is initiated after receipt of the 7th digit to determine whether more than seven digits are to be received. If an 8th digit is received before time out occurs, it is assumed that the foreign area code was intended and that a total of 10 digits will be received. If a time out occurs before the receipt of an 8th digit, it is assumed that the office code was intended and that all digits have been received and control is returned to the call connections program.

2.2.2.2 *MF Digit Analysis.* The multifrequency digit analysis program performs much the same analysis of the called digits but with a slightly different procedure. In transmitting the called digits by multifrequency pulsing, the local office sends a KP pulse at the beginning and a ST pulse at the end. Therefore, it is possible to wait until all digits are received and then perform the appropriate analysis, thereby avoiding in the case of conflicts the necessity to time for receipt of digits. Thus, when all digits have been received, the first three digits will again be analyzed, as in the case of dial pulse digits. However, if the first three digits are an office code, it is necessary only to check that seven called digits are received. In the case of a conflict, the resolution of the conflict is based again on the number of digits received. If a foreign area code were received, 10 digits are required. Although the originating office in this case must be a common control office and, therefore, should be screening the various illegal combinations that might be dialed by a customer, these checks are made as a safeguard against possible digit transmission failures which might have resulted in a partially valid check and have caused misrouting or failure in the toll network. Once a valid check of digits has been received, control is returned to the call connections program.

For multifrequency trunks the ST pulse that terminates called digit pulsing provides a traveling class mark. A discussion of the traveling class mark is given in the section describing ANI digit analysis.

For both dial pulse and multifrequency digits, the digit analysis programs are responsible for guarding against both invalid digit combinations and transmission failures. In the case where an invalid sequence of digits is received, as in a customer dialing error, control will be transferred to a program responsible for supplying an announcement to the customer informing him of this. In the case where it is apparent that a transmission failure has occurred, one of two courses is taken. For multifrequency trunks not carrying dial zero traffic and for all dial pulse trunks, the call will be routed to reorder. For multifrequency trunks where the trunk group can carry dial zero traffic, the call will be marked as dial zero and forwarded to the call connections program for operator handling as a dial zero call. The latter course of action is a safeguard against cutting off an originating office in the event of a wholesale failure of multifrequency senders in that office.

Both programs are also responsible for detecting calling party abandon during digit reception and transferring control to the appropriate call connections routine. Call connections will remove tbe

receiver and then transfer control to the disconnect program for idling of the trunk.

2.2.2.3 *ANI Digit Analysis.* The ANI digit analysis program is responsible for reception of the ANI digits from the local office. These digits identify the calling subscriber's line number for purposes of billing. No analysis, per se, is done on the calling subscriber digits themselves. They are merely recorded in the register associated with the trunk. However, certain information is transmitted to the TSPS through an information digit which is a part of the ANI digit transmission. The information digit informs TSPS of such things as identification failures in the local office, the fact that the originating line is a multiparty line and identification cannot be made, and whether or not the call has been locally service observed. In the case where the information digit indicates that identification was not made for either reason, the trunk register is marked so that when an operator is seized she will be informed that she must obtain the calling number and key this information into the system before advancing the call.

The ST pulse of the ANI transmission is used as a traveling class mark for dial pulse trunks carrying a mixture of one plus (station to station) and zero plus (operator assistance) traffic. Since the initial digit zero or one which is dialed by the subscriber is never seen by TSPS, the traveling class mark is used to indicate which digit was dialed. Also, for the case of trunks serving both coin and noncoin lines, the local office must forward information as to what type of line is being served on each call. This information is provided through one of four different ST pulses and is recorded by the ANI digit analysis programs in the trunk register. After receipt of the ST pulse, the ANI digit analysis program deactivates digit scanning and returns control to the call connections program. In the event of a transmission failure during ANI digit reception, the ANI digit analysis program will mark the trunk register to indicate an ANI failure, and then transfer control to the call connections program.

2.2.3 *Operator Actions Programs*

The operator actions programs are those which are responsible for a call while it is associated with an operator position. They may be divided into four groups, each having a separate function: (*i*) the key actions subroutines that are responsible for actions taken in response to a key operation on the position, (*ii*) the supervision program which is responsible for administering supervisory reports.

(*iii*) the monitoring programs which accomplish the duplication at the monitoring position of signals sent to a position being monitored, and (*iv*) the supervisor programs which handle the connections of a position to a supervisor console.

2.2.3.1 *Key Action Subroutines.* The key action subroutines handle all reports of keys operated at the position and must interpret these and respond appropriately to them depending upon the status of the call at the time the report is received. These programs are associated directly with the processing of a call. The strategy by which these programs are designed is based largely on tbe fact that tbere are few restrictions placed on the operator witb regard to tbe sequence in which she may supply information for the handling of a call. For example, while she must provide all data necessary for handling and billing of the call before releasing the call, the order in whicb she supplies this data is not restricted. On a call in which she might have to supply the calling number, a class of charge and a billing number, tbe order of supplying these items may largely be determined by tbe situation she encounters. For example, a customer might announce he wants to make a credit card call and give his credit card number and then announce that it is to be person-to-person or vice versa. Therefore, each key on the position has its own program, the entry into which is independent of what has passed before. The logic of the program consists of determining what has passed before by interrogating data in the software register associated with the position. Eacb key operation, as it is received and acted upon, will set one or more status bits or bytes so that a record of status is kept. In this manner, each key program can determine exactly what actions are to be taken. For example, on a dial 0 coin call, coin rating will take place when botb the called number and the class of charge have been received. Depending on which is received last, it would be either the start key (denoting end of called number keying) or the class of charge key whicb would obtain the rating. Both of these programs interrogate tbe coin rating status bits of the register to determine if rating has already been accomplished. These same coin rating status bits are set when a customer dialed call is received and rated prior to obtaining a position, tbus making the checks in these programs actually independent of the type of call which originally reached tbe position.

It is also possible that the called number required manual rating either on a dial 0 or 0+ basis. In this event, an additional piece of information is required from the operator and that is the rate treatment number which she acquires from her rate schedules or from a

rate and route operator. The ST key program which terminates the rate treatment keying would then initiate the coin rating. In a similar manner, all the keys use the data recorded in the position register to determine their ultimate course of action and in this way are independent of the sequence in which the operator handles the call.

The various lamps on the position are used to communicate responses to the operator from the system. For example, a lamp under a key is usually lighted in response to a key action when it is accepted. The most common indication that the system has not accepted a key is either to flash the lamp or not light it at all. The flashing lamp usually indicates that the system is in a state to accept keys, but this particular key action is out of sequence or cannot be accepted for lack of data that should precede it. For example, if the operator were to depress the ST timing key without first having a class of charge, the ST timing key lamp is flashed to indicate missing data. A key lamp that is not lighted after a key operation is usually an indication that the system is unable to accept any keys at that time, usually because it is still acting on some previous key action. Certain keys also cause indications to be given on the operator's numerical display panel. Examples of numerical displays given to the operator are: charge and minutes on a coin call, a time display in response to the time key, and a called digit number display in response to the called number display key.

2.2.3.2 *Supervision Routines.* The supervision routines administer reports of changes in the switch hook state of the calling and called parties during the time that a position is associated with a call. The routines control the state of two lamps on the operators console that reflect the supervisory state of the customers and record information in the position register that indicates the state of these lamps. Under certain circumstances, the routines also record connect time in the trunk register.

When the call is active at the position (not placed in hold), reports of changes in supervision from the calling party are received by the supervision routines and they cause the state of the supervisory lamp to follow these changes. When the initial called party off-hook report is received during the establishment of the call, the supervision program initiates timing to determine if the off-hook is a true answer or the beginning of a busy or reorder. If a true answer is determined, the supervision routine extinguishes the called party lamp and records

the answer in the position register. If a busy or reorder is detected, the supervisory lamp remains lit, but the operator is able to hear the busy tone from the called party direction. After a true called party answer has been determined, control of the supervisory lamp is provided in the same way as for the calling party.

If the call is placed in hold by tbe operator, the initial off-hook report from the called party is handled as described above. A busy or reorder signal in this case would be heard by the calling party only with no change in state of the called party's supervisory lamp. Subsequent to a true called party answer tbe supervision routines would control the state of the supervisory lamp as described above.

With a held call in a position the supervision routines will initiate timing if an on-hook report is received from the calling customer. If a steady on-hook is detected, the supervisory routines will light the supervisory lamp steady. If flashing is received from the calling party, the supervisory lamp will be flashed at a fixed rate until further customer action is determined.

2.2.3.3 *Monitoring Programs.* The work of an operator at a position can be observed at a second position by the duplication of tbe lamp indications being sent in response to the actions of the operator handling the call. A special position equipped for this function is called the monitor position, and the monitoring program administers the duplication of the signals. While the displays at tbe monitor position are dependent on actions taken at the monitored position, the program is designed so as to be effectively independent and noninterfering with the programs actually handling tbe call.

The monitor position is activated by means of a key operated switch at the monitor position. In response to operation of this switch, the monitoring program displays the monitoring position number on the numerical display. This is an indication to the monitoring operator that her signal has been received and the program is now prepared to monitor whichever position she specifies. She operates two digit keys denoting the position number which she wants to monitor. The monitor program then checks tbe position register associated with the monitored position and updates the lamps on the monitor position to the present status of those at the monitored position. It also sets a bit in the monitored position register indicating that that position is being monitored. Subsequent signals sent to the monitored position are then passed to the monitoring position as a result of the

key action programs checking this monitoring bit. The monitoring routines return control to the key action routines in such a way that, to those programs, it is as if monitoring were not even taking place. Thus, the monitor program does not interact with the key operations nor in any way affects the processing of a call.

Once having established a monitor link with a given position, the monitor may observe as many calls from that position as she likes. If she desires to monitor another position, she merely operates her POS RLS key and keys in another two digits. If there were any displays at the monitor position pertaining to the previously monitored call at the time of POS RLS, they will be extinguished, and with the reception of the two digits the linkages will be reestablished to the new position as previously indicated. The only key actions which are accepted from the monitor position while it is in the monitoring mode are the digit keys as indicated above and the position release key. Any other keys which might be depressed while the position is in the monitoring mode are ignored by the system.

2.2.3.4 *Supervisor's Console Program.* TSPS provides for any operator to contact her supervisor or service assistant for assistance at any time by means of a supervisor (SR) key at her console. The service assistant answers such calls or may originate calls to any position in that group or to other selected points from a call-director-like console. Monitoring of the operator's voice connection is also provided from the supervisor's console.

Operation of the SR key at the operator's position causes a key lamp to flash at the supervisor's console alerting her to the waiting call. Depressing the key at the console completes a connection to the operator and steadies the flashing lamp. If the operator has a customer call in access, the supervisor's connection is bridged on the call, allowing a three-way conversation. Release from the connection can be effected by either the operator or service assistant.

The service assistant can originate a call to a position for talking or monitoring by depressing the talk key or monitoring key respectively and then keying a two-digit number to identify the position. For a talking connection the SR key lamp is flashed at the desired position under program control. The operator responds by depressing the SR key, at which time the lamp is steadied and the connection established. On a monitoring connection, a low-loss bridge is established on the operator's talking circuit with no indication to the operator. Release

from the monitoring connection is effected by reoperation of the monitor key at the supervisor's console.

2.2.4 *Disconnect Program*

2.2.4.1 *Call Talking States.* The processing of a call after it has been released from a position for the first time is determined in part by the call state that is established for the call at the time it is released from the position. After initial position seizure, the call state is determined by the actions taken by the operator and the states of the calling and called parties at the time the call is released from the position. The four major states are: (*i*) noncoin notify, (*ii*) coin paid, (*iii*) residual call, and (*iv*) general.

The noncoin notify state is established upon a customer's request to be notified at the end of a prescribed interval chosen by the customer. The operator sets the notify state by operating the KP NFY key and keying in the notify interval. After position release and called party answer, the call is entered into a timing list, and is subsequently handled by the disconnect program.

The coin paid state requires that a timing function be initiated at the time of position release. In this case the interval is established to cause the call to be returned to the disconnect program to collect the initial period coin deposit.

The residual call talking state is established for certain calls such as mobile, marine, and overseas calls that require connection to a cord board operator via the toll switching system serving the TSPS. Control over these calls lies with the cord board operator and a TSPS operator is never brought back on the connection.

With the exception of the residual type call, calls that are subsequently reconnected to a position after being released, retain their call state identity unless the operator takes some special action during the reconnection. Special action might be initiated by a customer's request to terminate or cancel the call and begin a new one. This situation can be handled by changing the call state identifier and treating this connection to the operator as an initial position seizure. Further action on the call is taken as if the call had reached the position as a new seizure.

The general state applies to calls which do not fit into one of the other three states. In this state the initial actions of the disconnect program are common for all calls in this state. However, as will be seen

later, there are some variations in the disconnect actions that apply to this state.

Once a call has reached the talking state, it is under the control of the disconnect program. Calls in this state are divided into the four major classes described above. These four classes of calls constitute the major legs of the disconnect program. A description of the actions performed by the disconnect programs for these calls is given below.

When a notify timeout is received on a noncoin notify call, the disconnect program obtains a position and, having established the connection to the position, transfers control to the operator programs once again at a point which signifies that this is a notify seizure. Should the customer disconnect prior to the timeout, the disconnect program proceeds to take normal disconnect actions in recording the call and releasing the trunk. If the customer flashes to recall an operator, the disconnect program establishes a connection to the operator. However, the operator programs are entered at a point which signifies a flashing recall and not a notify timeout. Also, certain data is recorded in the auxiliary register, which allows the operator programs to resume the notify timing at the point at which the flash occurred.

For coin calls, the disconnect program is responsible for handling the various initial period and overtime interval timeouts. Initially, the call is placed in a timing list whenever answer has been established, or upon release from the position, which provides a timeout 18 seconds before the end of the initial period. When this timeout occurs, the disconnect program establishes a connection to a coin control circuit and transfers control to the coin control programs which will execute a coin collect sequence and then return control to the disconnect program. At the conclusion of the coin collect sequence the call is returned to the timing list for the remainder of the initial period. At the end of the initial period, the disconnect program establishes a connection to an operator and transfers control to the operator programs at a point which signifies a coin notify seizure. Upon release from the position after the coin notify seizure, the call is placed in a timing list for a grace period of 6 seconds. If the customer disconnects before this grace period is up, the call will be disconnected with no further charging. If a grace period timeout occurs before disconnect, overtime timing is continued. The disconnect program receives timeout returns from the timing program for each ensuing overtime interval. Each time such a timeout occurs, a counter

in the auxiliary register is incremented to keep track of the number of overtime intervals for charging purposes. A maximum of 10 overtime intervals can pass before any system action is taken. When the 10th overtime interval timeout occurs, a position will be seized and a charge due seizure indicated. If the customer again remains off-hook after the position releases, a similar timing cycle is started again. This will be repeated for each 10 overtime intervals as long as both parties remain in the off-hook state. During any of these overtime cycles, if the customer disconnects or flashes, indicating to the system that he has completed his call, a position will be seized for a charge due request. However, in this case the forward connection is released as it is assumed that the call has been terminated. When the operator releases the call, control is transfered from the operator programs to a point in the disconnect program which will record the billing information and restore the trunk to the idle state.

For a call in the residual call state, disconnect actions are under joint control of the operator and the calling customer. The disconnect program, upon receipt of a flash or a disconnect from the calling customer, does not take disconnect actions but relays the signal to the cord board operator and waits an indication from her as to the disposition of the call. Should the called end go on-hook at this time, indicating that the cord board operator has released the connection, the disconnect program proceeds with the usual disconnect action. However, if a wink is received, the disconnect program attaches an MF receiver to the outgoing side of the trunk to receive inband signals which must then be repeated via a coin control circuit to the local office. The signal could be coin return, coin collect, or ringback. If an on-hook is received from the outgoing side of the trunk first, the disconnect program waits to receive a disconnect from the incoming side of the trunk before taking disconnect action.

All other calls in a talking state not specifically of the types already discussed are administered by the general talking state leg of the program. If an on-hook is received from tbe calling side of the trunk, flash timing will always be performed to discriminate between flashes and true disconnects. If a disconnect is established, a billing record will be made, the trunk connections released, and the trunk restored to the idle state. If a flash is detected, the disconnect program cbecks the talking state index to see if this type of call has flash recall allowed. If it has, a connection is established to a position and control returned to a point in the operator program which indicates a flashing

recall. If flashing recall is not allowed on the call, the flash is ignored and the call returned to a talking state. At the time a disconnect on a call in this category is detected, the disconnect program also determines if it is necessary to collect or return a coin. If collect or return is required, it will perfrom the function before completing the other disconnect actions.

#### 2.2.5 *Other Call Processing Programs*

There are several additional call processing programs which do not directly control the processing of a call but which are significant attributes of TSPS No. 1. Among these are the coin rating and computing programs and the service observing program. A brief description of these programs is included here for completeness.

2.2.5.1 *Coin Charging Programs.* One of the types of calls the TSPS No. 1 must handle is the coin paid call. A major advantage of TSPS over cord boards is that the initial period and overtime charges on these calls can be automatically computed and displayed to the operator, saving her time and effort and providing a more accurate result than humanly possible. These functions are performed by two programs, the coin rating program and the coin charge computing program.

The cost of a coin call is based on several factors: (*i*) the appropriate rate schedule, (*ii*) person or station class of charge, (*iii*) distance, (*iv*) time of day, and (*v*) possible reduction of rates for certain holidays or days of the week. The rate schedules are a means of dividing the rates which might be applicable to a call into categories based on the relation of the originating point to the terminating point. Calls which originate and terminate in the same state are subject to intrastate schedules. Calls which originate in one state and terminate in another are subject to an interstate schedule. Calls which originate within the continental boundary of the United States and terminate outside the continental boundary are subject to international schedules. TSPS No. 1 allows for up to five schedules to be rated in any given installation in any combination of the following:

(*i*) one interstate schedule,
(*ii*) canadian schedule (international),
(*iii*) mexican schedule (international), and
(*iv*) three intrastate schedules.

A numerical index has been established which defines the charges applicable to a call, accounting for the aforementioned variables.

This index is referred to as the rate treatment number (RTN). The RTN can be broken down into three components: (*i*) that portion which defines the applicable schedule, (*ii*) that portion which indicates that person or station rates apply, and (*iii*) that portion referred to as the rate treatment index (RTI), which is an index into a table defined by (*i*) and (*ii*) containing information in the form of charges and minutes for the initial and overtime periods. The process of rating a call, then, is one of determining the proper RTN.

The rate schedule which applies is determined by interrogating memory relating to the originating and terminating points and arriving at one of the schedules defined above.

The person or station status is indicated to the coin rating programs by the program requesting the rating. In the case of customer dialed calls where rating is done prior to seizing a position, the class of charge is inferred from the prefix dialed by the customer. If a 0+ call has been dialed, it is assumed that person rates apply. If a 1+ call has been dialed, it is assumed that station rates apply. Requests that are generated while a call is connected to a position use the class of charge entered by the operator to establish this status.

In order to derive the RTI, a rate line is first determined. A rate line defines a set of rates applicable to a call and is based on the distance between originating and terminating points. The way in which this item is obtained varies depending on how the numbering plan area (NPA) containing the terminating central office (TCO) is rated.

To economize on memory requirements for coin rating, a given TSPS installation may not rate all NPAs. In general, those NPAs which receive the largest volume of traffic will be rated. If the NPA is not automatically rated, the call will be taken to the operator with a manual rate indication. The operator will then obtain the RTN via her bulletins or from a rate and route operator and key it into the system.

An NPA which is rated falls into one of two categories, a single rate area or a multirate area. A single rate area is one such that a single rate line applies to all calls terminating in it from the TSPS installation in question. If this is the case, the rate line is determined immediately. For multirate areas the rate line must be determined based on the TCO code and the originating NXX code. The TCO code itself can fall into one of four categories: (*i*) vacant (probable dialing error), (*ii*) manually rated, (*iii*) V and H ratable, and (*iv*) exception. The first case would be routed to reorder if it were customer dialed, and the second would be taken to an operator with a manual

rate indication as before. The last two will be described briefly below.

The V and H system is a coordinate system superimposed on the United States, Canada, and Mexico. The V stands for the vertical coordinate and the H stands for the horizontal coordinate. By using the location of the originating and terminating points specified in the system, a measure of the airline distance between the points is calculated. This measure is then used to examine a table (for the schedule which applies) to obtain the rate line.

For the exception case, the rate line is determined by a direct look up method. Instead of finding V and H vector data, an address is found of a location which contains the rate line.

Having obtained the rate line by one of the above methods, the RTI can then be calculated. This is accomplished by applying the remaining variables of time of day, holiday or day of week considerations. These variables are determined through a series of table look ups. The three components of the RTN are then determined and the initial period and overtime charges can be determined. In the case of a manually rated call, the operator supplied RTN is identical to what would have been obtained had the call been automatically rated. In either case, the RTN is stored in the auxiliary register for subsequent use.

To obtain the initial period (IP) charge and minutes, the RTN is used to obtain data from a table for the specified rate schedule. This data consists of: (*i*) the IP charge after tax and rounding, (*ii*) timing information, (*iii*) the IP charge before tax and rounding, and (*iv*) indicators as to what tax rates apply. The first two items are used to provide the display to the operator. The first three items are stored in the auxiliary register to be used as a basis for subsequent calculations if the call goes into overtime.

When an overtime calculation is to be made, the coin computing program retrieves the RTN from the auxiliary register and, again the same data table for the specified rate schedule is read. This time, however, the overtime charge is obtained. Using the count of elapsed overtime intervals that are supplied by the client program, the overtime charges are calculated and returned to the client.

In performing the rating and computing functions, the coin programs provide considerable flexibility in allowing for a variety of federal, state, and municipal tax structures, with the calculated amounts rounded to the nearest nickel. Considerable flexibility is also allowed in making changes in coin rating and computing data. The addition

of a new office in an area rated by a TSPS installation requires that new data be added in memory. A rate change also affects the contents of memory depending on the extent of the change. These and other modifications to coin data can be made by means of recent change techniques or via a magnetic tape that can be prepared for massive changes well in advance of their introduction. (See Section VII, Recent Changes, and Section VIII, Program Tape Unit Control.) Hence, the operating company can introduce changes in coin rating with ease.

2.2.5.2 *Service Observing Program.* The function of a system such as TSPS No. 1 is, of course, to provide a good quality of service to the telephone customer. Therefore, it is necessary to provide a means for determining that the service being provided is acceptable by the standards which have been set for it. One of the means by which the service can be measured is through service observing. This is a feature whereby botb tbe system and the operators are monitored to see if the standards are being met.

The service observing function in TSPS has been implemented so as to allow observing on any trunk in the system and any operator connection which is established to that trunk. At a point in the call connections program just prior to seeking an idle position to serve a call, control is transferred to the service observing program. At this point, the service observing program performs a directed scan of a control ferrod to establish whether service observing is in effect for the system at that time. If it is, a second check is made to see if the class of call which is being handled is a candidate for observing. If either of these checks are negative, control is returned to call connections program and the call is handled in the normal manner. If, however, service observing is to be made on this call, the service observing program will then secure the next idle TSPS position. At that time, a third check is made to see if that particular chief operator group is being observed. If it is not, control is again returned to the call connections program and the call handled in the normal manner. If all the checks pass, the call is to be observed and the service observing program will reserve a path to the operator. However, instead of connecting directly from the trunk to the operator cut-through circuit, it will connect via a service observing monitor circuit which has a double appearance on the network. The circuit has one appearance on the trunk side and one on the position side as shown in Fig. 1. At this time, it sends a seizure signal via the monitor circuit which causes

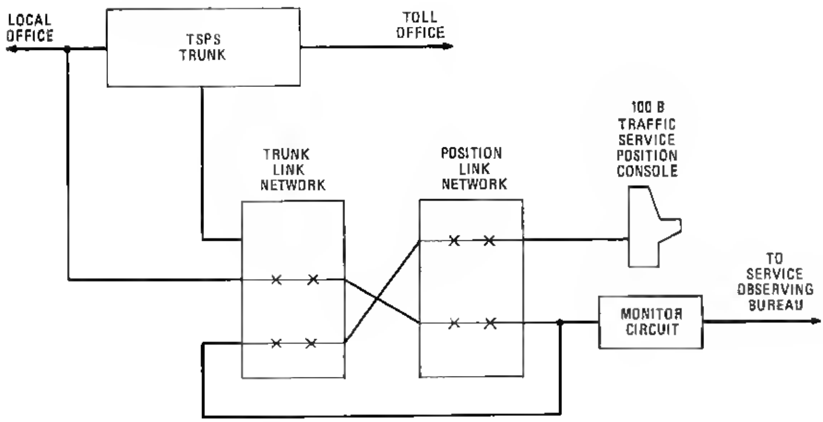


Fig. 1—Insertion of service observing monitor circuit into position connection.

any idle service observing desk to be associated with this monitor circuit. When this connection is successfully established, information relating to the call such as calling and called number, chief operator group and position number, etc., is transmitted to the service observing desk. When this initial data transmission is complete, the connection to the operator is established and control transferred to the operator programs at the initial position seizure point. The trunk register associated with this call has been marked to indicate that the call is being service observed. All subsequent actions on the call for which indications must be sent to the service observer check for this mark and, finding it set, transfer control temporarily to the service observing program which transmits data to the service observer. Control is returned to the programs at a point such that they complete their functions as though the service observer were not attached. Thus, interfacing with service observing is accomplished with no interference to the normal processing of the call except for the delay in sending the initial set of data to the TSPS position.

The service observer may release herself from the connection at any time. When the service observing program detects the release signal from the observer, it releases the service observing desk and administers the network connection to the monitor circuit according to the state in which the call is established. If the call is still connected to a position, for example, the connection to the monitor circuit cannot be removed since this would disrupt the connection to the operator. In

this case, the service observing program marks a bit in the trunk register indicating that the service observer is no longer attached. Upon position release, the position release key program then releases all the connections associated with the service observing monitoring circuit. If the service observer is still connected at the time of position release, the position release key program recognizes this condition and releases the operator cut-through circuit connection but retains the connection through the network to the service observing monitor circuit. If the service observer release signal is received when the call is no longer connected to a position, the service observing program releases the connection to the monitor circuit and erases the indications in the trunk register indicating that tbe call is being observed. Further handling of the call after this proceeds without any interfacing with service observing.

## III. TEMPORARY MEMORY

During the processing of a call, numerous data is transmitted to and from the various input-output and call control programs. This data is kept in temporary memory associated with the various call processing or service routines. The recording of information in temporary memory is the means by which the various parts of a call are linked together and continuity is maintained.

### 3.1 *Input-Output Oriented Memory*

Similar to ESS No. 1, each scan point of a trunk has two bits of temporary memory associated with it. These two bits are used to indicate: (*i*) that the facility is idle and may originate a request for service; (*ii*) that the trunk is in the talking state and being monitored for disconnect; or (*iii*) that the scan point state is being monitored by another means temporarily, and that the supervisory trunk scan should disregard any changes that occur.

The receiver scanning program has several blocks of temporary memory to administer scanning of receivers for digits. One block is used to store signal present indications from the receivers. Another block contains receiver activity information that indicates that a receiver is connected to a trunk and ready to receive digits.

Another register associated with input programs is the timed scan junior register which is a block of memory used in timing on-hooks to filter out hits from true disconnects. It is also used in performing timing to detect calling party flashes and called party disconnects.

A third register, tbe flash scan timing register, performs a similar function for the called party side of the trunk to distinguish between answer and off-hook flashing, indicating the return of a busy, reorder, or circuit busy signal from the toll end of the trunk.

### 3.2 *Hoppers*

As was mentioned in tbe description of the input-output programs in Section 2.1, input data to the system is obtained during H- and J-level interrupts and is analyzed by ground level programs. Hoppers are used to buffer the data between the input programs and their base level processing programs with the data served on a first-in—first-out basis. Each input program has one or more hoppers by which it relays information to base level, and each entry contains some identification that associates the data with a particular call. The data and its identification varies with the function performed by the program. For example, the supervisory trunk scan program reports an off-hook in the trunk seizure and answer hooper by loading the trunk scan number which identifies the trunk on which the change of state has occurred. In the case of digit scanning the trunk register address is loaded in the digit hopper along with the digit received.

### 3.3 *Output Buffers*

Data that is generated by base level programs for execution or transmission by interrupt level programs to the peripheral system requires output buffers. Peripheral order buffers (POBs) are the principal means by which information is buffered before being sent to units such as networks and signal distributors. Eacb POB is a fixed length table with space reserved for call identification, an address to which control is to be returned after the data has been transmitted, and tbe data itself. Tbe number of these buffers is traffic dependent and must be engineered according to the needs of each office.

Similar information that is sent to the position subsystem is buffered by position information buffers (PIBs). The arrangement of the PIB is similar to tbat of the POB except that PIBs are designed to store orders whose format is peculiar to the requirements of the position subsystem. PIBs must also be engineered for eacb office. Otber buffer areas include those for buffering digits to be outpulsed, TTY characters to be sent to teletypewriters, and call billing information destined for the AMA tape but unlike PIBs and POBs are not engineered items.

### 3.4 *Call Control Registers*

Call control registers are blocks of unprotected memory that are used to record transient information about a call at various stages in the handling of the call. As was pointed out in the introduction, TSPS differs from ESS No. 1 in the provision of these registers. In general, the TSPS registers are dedicated to the circuit with which they are associated in the processing of a call. The format of these registers is standardized although their size and information content may vary, and to conserve memory some areas of registers may serve different functions at different phases in the call.

The standard format of these registers defines the first five words of all call registers. Included in the information stored in these words are the paramater register address, a link word used to link other service registers, two queue words to link the call register to various link lists, and a call state identifier word to define the state of the call. Additional space may be required in some call registers for more information, and this space uses standard layouts as much as possible.

Briefly, the three major call registers are the trunk register, position register, and coin control trunk register. The trunk register is used to store control data and all of the billing information with the exception of coin charging information, credit card number and charge to third number. The position register stores control data on all calls in access or hold on the associated position. This data includes memory of what keys have been operated and what lamps lit. The third register is the coin control trunk register and is associated with the coin control circuit. Unlike other service circuits, this circuit was given a call register to vest control of the coin actions (collect, return, and ringback) in one program.

### 3.5 *Service Registers*

In addition to the call registers described above there are a number of registers associated with service circuits or service routines used in processing a call. Service registers provide dedicated memory for MF receivers, DP receivers, outpulsers and the like with a standard format employed in the first three words. In these words is stored the address of the associated parameter register, a link word to connect the service register with the call register, and several status bits related to maintenance states. Again, some registers such as the MF outpulser register require added memory for storing additional information.

Other service registers associated with certain program functions include the time scan junior register and the flash scan junior register, the path memory annex, and the auxiliary trunk register. The time scan and flash scan registers are similar to those in ESS No. 1 and are used for detecting answers, disconnects, hits and flashes. The path memory annex is linked to a call register during a network connection to store the identity of the connected circuits, the paths used, and their status. Finally, the auxiliary register is linked to the trunk register to provide additional storage for coin paid charges, the third party number on charge to third party calls, and credit card and other special billing information. The numbers of these service registers are engineered according to the needs of each office.

## IV. PROCESSING A TYPICAL CALL

The following is a description of a typical 0+ coin call from a Step-by-Step office which illustrates many of the actions that take place in handling a call with TSPS.

### 4.1 *Detection of Origination*

Figure 2 portrays the memory and program associated with the detection and processing of an origination. The supervisory trunk scan program detects a change of state in the ferrod of the incoming side of the trunk and enters the trunk scan number into the trunk seizure and answer hopper. The trunk scan program scans MF trunks at a 200 millisecond rate and DP trunks at a 100 millisecond rate. Thus, detection of an origination is at most 200 milliseconds after

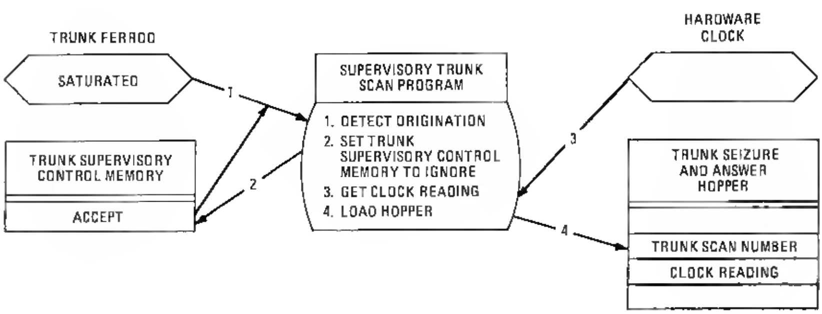


Fig. 2—Detection of origination.

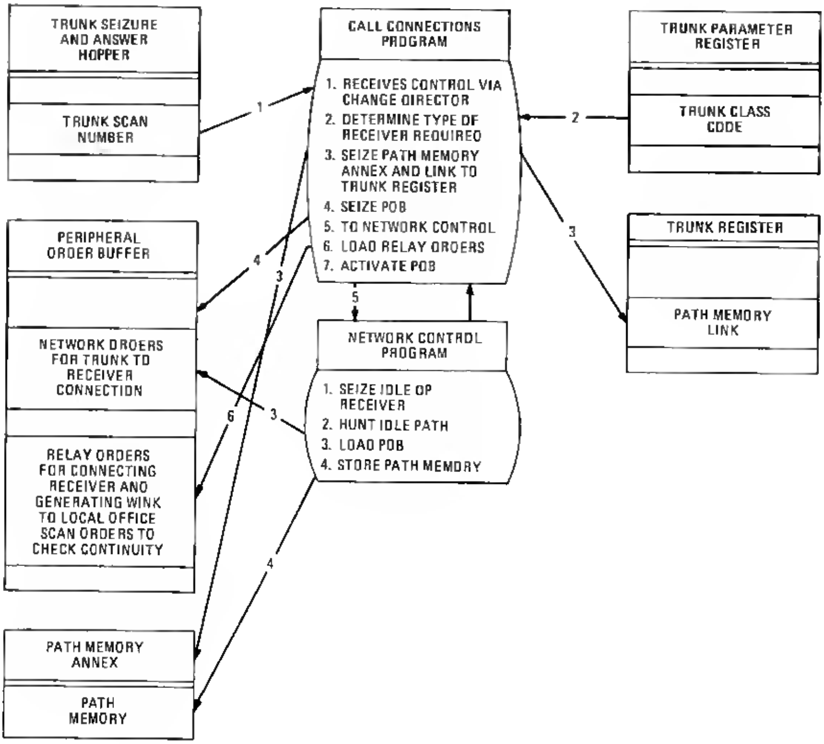


Fig. 3—Initial actions for connection of trunk to digit receiver.

the time the trunk is seized for the MF case and 100 milliseconds for the DP case.

### 4.2 *Connection of a Trunk to a Digit Receiver*

The initial actions of the call connections program are illustrated in Fig. 3. The change director program, entered periodically from executive control, unloads the trunk seizure and answer hopper and via translation determines that the scan point is associated with an incoming trunk. It thereupon executes a transfer via the state word of the trunk register. The state word of the trunk register directs transfer of control to the originating portion of the call connections program. At this point the call connections program interrogates the parameter register associated with the trunk to determine the various pieces of information necessary for connecting the appropriate type

of receiver. The call connection program then seizes an idle POB and a path memory annex. Control then transfers to the network control program along with information defining the type of receiver to be connected and identifying this particular trunk. The network control program loads the POB with orders to connect the trunk and the receiver and records the corresponding linkage data in the path memory annex. The call connections program also loads the POB with the appropriate relay and scan orders associated with establishing a connection to the receiver. After loading orders in the POB, the call connection program then activates the POB. As shown in Fig. 4, the POB execution program causes the orders in the POB to be executed. After successful execution of the orders loaded in the POB, the connection as shown in Fig. 5 is established. Also, the relay orders in addition to establishing the facility connection as shown generate a signal to the local office that the TSPS is ready to receive digits. Upon completion of the execution of the POB (as shown in Fig. 4), control is returned to the call connections program which then idles the POB and activates the receiver by setting a bit in the receiver scan activity control memory. This causes the digit scanning program to detect and transmit digits as they are received via the digit hopper to the digit analysis program.

### 4.3 *Digit Analysis*

#### 4.3.1 *Reception of Digits*

As shown in Fig. 6, the receiver scan program detects the appropriate change of state in the digit present ferrod of a receiver denoting that a digit is present. It then scans the digit ferrods of the receiver and stores the digit along with the trunk register address for this trunk in the digit hopper.

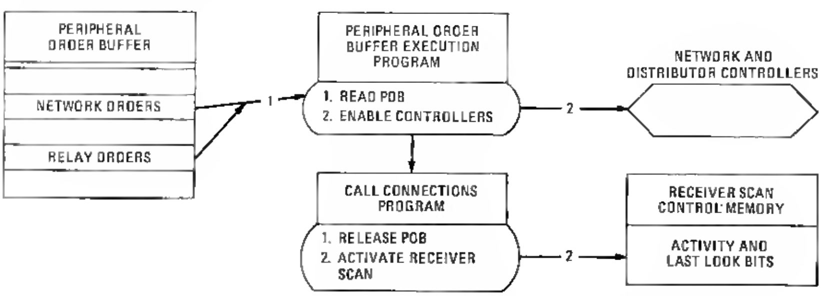


Fig. 4—Actions for connection of trunk to receiver.

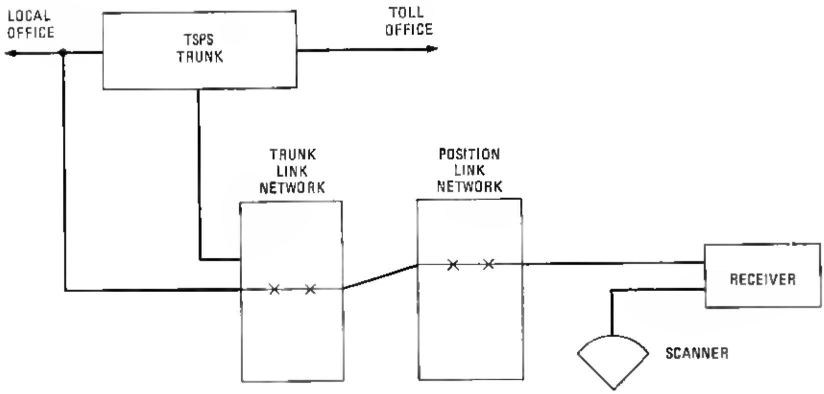


Fig. 5—Receiver connection.

The digit analysis program, which is shown in Fig. 7 and which is scheduled by an executive control program, treats the entry in the hopper, stores the digit in the trunk register, and increments the digit reception counter. When the digit analysis program recognizes that the third digit has been received, it transmits these first three digits to the called digit translation program which returns information for the analysis of the dialed digits. The type of information returned here defines the first three digits as being an area code, an office code, an invalid code, etc. For our example it will be assumed that the digits are recognized as a foreign area code which means that 10 digits are ultimately to be received. For this case the digit analysis programs marks the trunk register to indicate that no further action is required until receipt of the 10th digit.

#### 4.3.2 *Completion of Digit Reception*

In the case of the 10 digit call (as shown in Fig. 8) when the digit analysis program determines that the 10th digit is received, it shuts off the receiver scan program for this particular receiver, marks the register as dialing complete, and returns control to the call connections program.

### 4.4 *Reception of Calling Party Identification*

#### 4.4.1 *Establishing the Receiver Connection*

Upon receipt of control from the digit analysis program, the call connections program again interrogates the parameter register of the

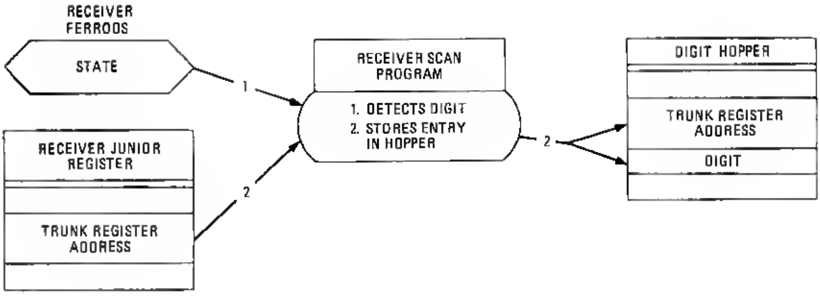


Fig. 6—Reception of digits.

trunk to determine if automatic number identification (ANI) is provided for this office. For our example it is assumed that the local office is an ANI office.

In this case, the call connections program seizes a peripheral order buffer and again transfers control to the network control program which loads orders to break the connection between the trunk and the dial pulse receiver and to establish a connection to a multifrequency receiver for reception of the ANI digits. These actions are shown in Fig. 8. At the completion of the loading of the POB the call connections program activates the POB.

After the POB execution program successfully completes the execution of the orders in the POB (Fig. 4), control is returned to the call connections program.

The successful execution of the POB actions establishes an identical connection as shown in Fig. 5 except an MF receiver is now connected. The relay actions in this POB also generate a signal to the local office to indicate that the TSPS is now ready to receive the ANI digits. The call connections program (as shown in Fig. 4) then idles the POB and activates the receiver scan program to scan for and transmit to the ANI digit analysis program via the digit hopper the digits as they are received.

#### 4.4.2 *Reception of ANI Digits*

As shown in Fig. 6 the receiver scan program again periodically interrogates a signal present ferrod of the receiver to detect when a digit is present. Upon detection of the appropriate change of state in the signal present ferrod of the receiver, the receiver scan program scans the tone ferrods of the receiver and stores this information along with the trunk register address in the digit hopper.

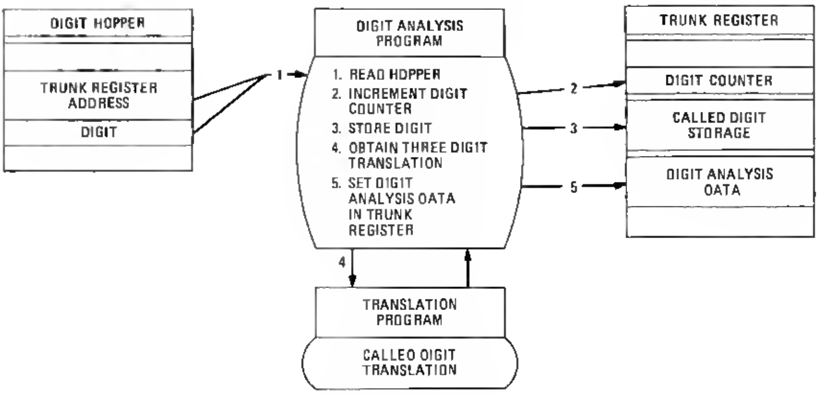


Fig. 7—Analysis of third digit.

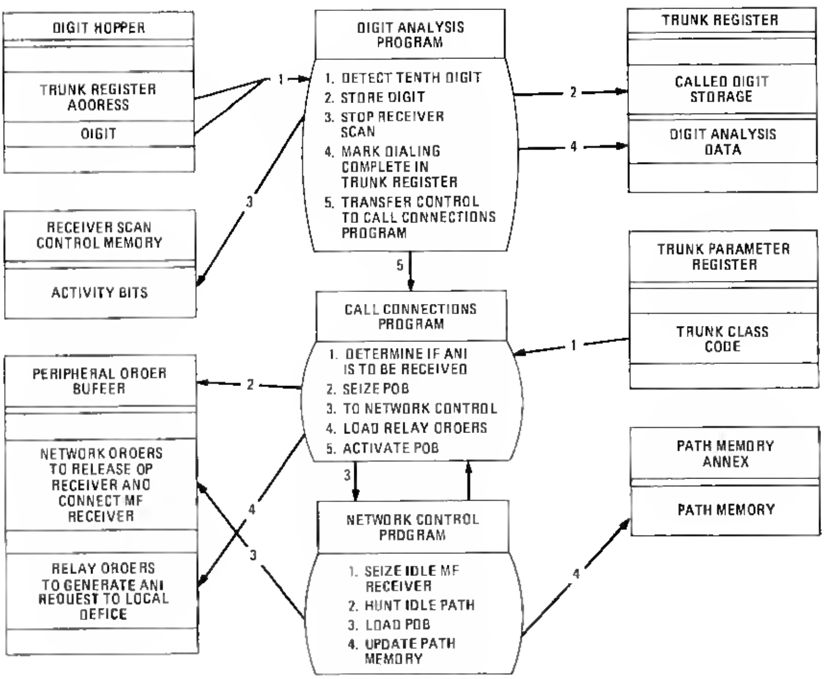


Fig. 8—Actions upon receipt of last digit.

The ANI digit analysis program obtains the ANI digits from the bopper and performs the appropriate analysis actions. The first digit which is received must be a KP pulse to insure that none of the first digits have been missed. This is not stored in the register, but the fact that it has been received is indicated in the control words used by the analysis program. The next digit received is the ANI information digit. This is analyzed and depending on its value certain information may be recorded in the trunk register. For example, the information digit may indicate that the call has been locally observed. If so, a service observed mark in the trunk register is set. This digit may also indicate that the local office is unable to identify the calling party due to an equipment failure or to the fact that the originating party has a multiparty line such that his number cannot be determined. In this case bits in the trunk register are set to indicate that operator identification must be requested and which type of indication is to be given to the operator. This would also indicate to the program that a 7-digit line number is not to be expected. In the example being used here of a dial pulse office, the last signal to be received, the start (ST) signal, is the traveling class mark. There are two possible ST signals that may be received—one indicating that the customer had prefixed the number he dialed by a one, the other indicating that he prefixed his number with a zero. In this example, we assume that the customer prefixed a zero. Upon receipt of the start signal, the receiver is deactivated and control is transferred to the call connections program.

### 4.5 *Establishing a Connection to a Position*

At this time the call connections program (as shown in Fig. 9) examines the data recorded in the trunk register and determines that a position should be attached to serve the call. Before proceeding to seize a position, the coin rating data is obtained and stored in the auxiliary register. It then seizes an idle POB and transfers control to the network control program which loads the POB with orders to establish a connection to both a position and an outpulsing circuit. If either of these circuits is unavailable, the call connections program queues until both circuits are available. The call connections program loads the orders to perform the appropriate relay operations required to complete the connection and alert the operator with a zip tone when she is connected. After completion of the loading, the call connections program then activates the POB.

As shown in Fig. 10, the POB execution program causes the orders

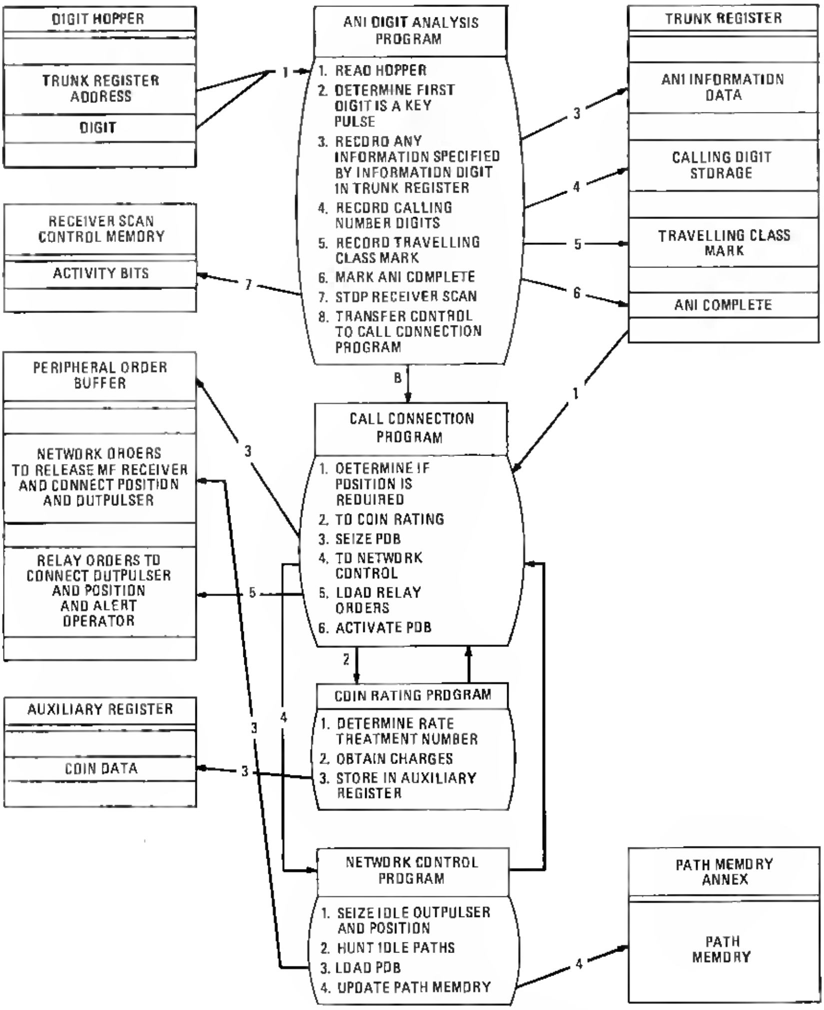


Fig. 9—Actions for receiving ANI digits and seizure of position and outpulser.

in the POB to be executed. The successful execution of these orders results in the configuration shown in Fig. 11, where the position is connected to the calling customer side of the trunk and the outpulser connected to the toll side of the trunk. Upon completion of the POB, control is returned to the call connections program. The call connection program idles the POB, transfers to the outpulser loading routine

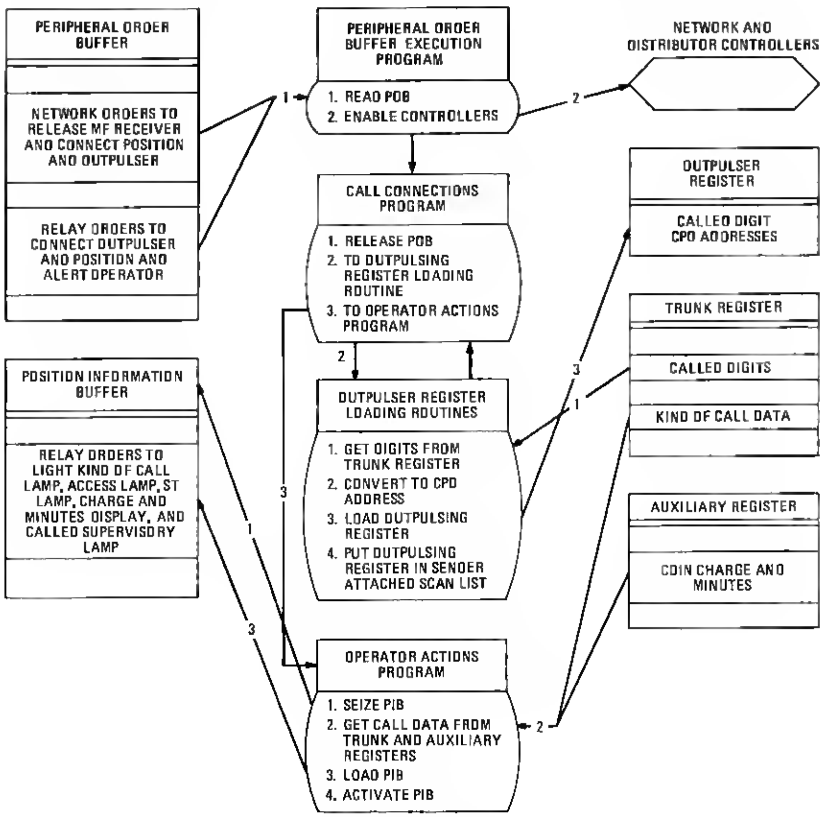


Fig. 10—Actions for connecting position and outpulser.

for loading of digits to be outpulsed, activates sender attached scanning for the outpulser, and transfers control to the operator actions program.

### 4.6 *Operator Actions*

#### 4.6.1 *Initial Display*

As shown in Fig. 10, the initial function of the operator actions program is to seize a position information buffer (PIB) and provide the operator with a display on her position which indicates the type of call that she is to handle. In this case the "0+ coin" kind of call lamp will be lighted. The access lamp for the loop which is to be used

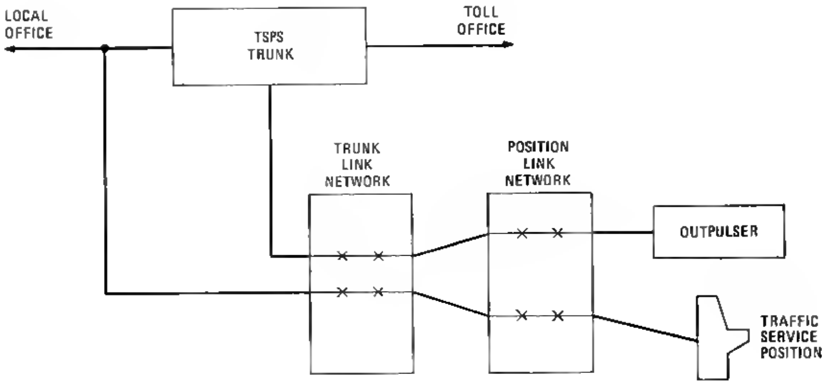


Fig. 11—Outpulser and position connection.

for the call and the ST lamp to indicate that an outpulser is attached are also lighted. Since this is a coin call, the initial display routines go to the auxiliary register where the call connections program stored the initial charge information it received from the coin rating programs and uses this data to generate a charge and minutes display on the position's numerical display panel. After completion of loading the PIB, the PIB is activated.

Upon successful execution of the PIB (as shown in Fig. 12), control is returned to the operator actions program which then idles the PIB. The next action taken by the operator actions program is dependent upon the sequence of keys operated by the operator. Other inputs to the operator actions program will be receipt of sender attached notification, followed by outpulsing complete notification, and called party answer. While these three stimuli follow each other in sequence, they

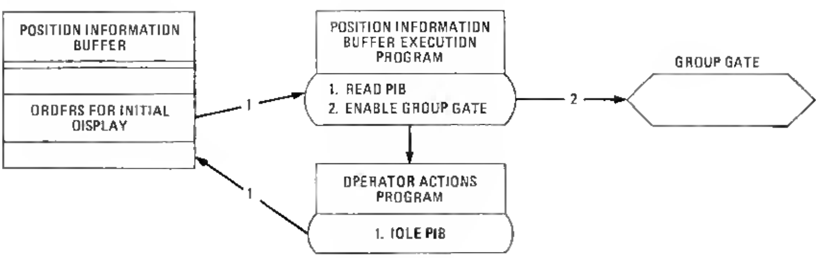


Fig. 12—Final actions for connecting position.

could be interspersed with certain key actions initiated by the operator. For sake of simplicity in this example a particular sequence of events will be assumed.

#### 4.6.2 *Receipt of Sender Attached*

As shown in Fig. 13, the sender attached scan program upon detection of sender attached at the toll office deactivates sender attached scanning and loads a report in the miscellaneous scan report hopper. Sometime later, when this report is unloaded from the hopper, control is transferred to the operator actions program. At this point (Fig. 13) the operator actions program activates outpulsing of the called number.

#### 4.6.3 *Outpulsing of the Called Number*

The outpulsing program is entered periodically and sends one digit at a time to the toll office until it determines that all digits have been sent. Upon completion of outpulsing the called number (shown in Fig. 14), the outpulsing program deactivates outpulsing for this outpulser and loads an outpulsing complete report in the miscellaneous scan report hopper. Sometime later when this report is unloaded from the hopper, control is returned to the operator actions program. At this time a POB is seized and control is transferred to the network control program to load orders to release the connection to the outpulser circuit. The operator actions program loads orders to close the

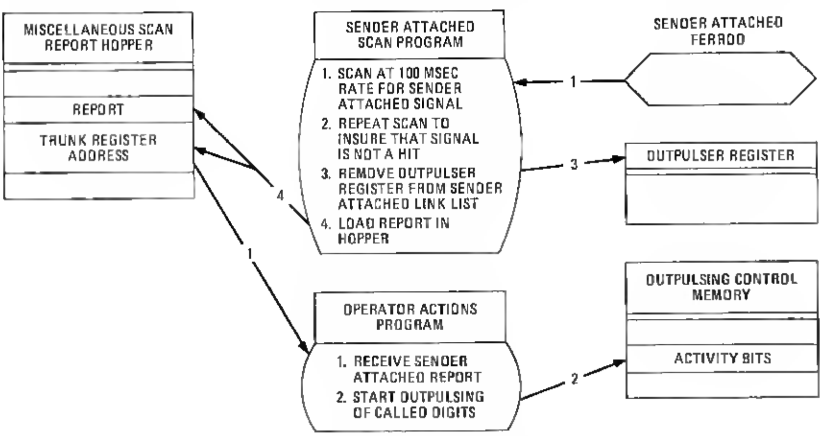


Fig. 13—Receipt of sender attached signal.

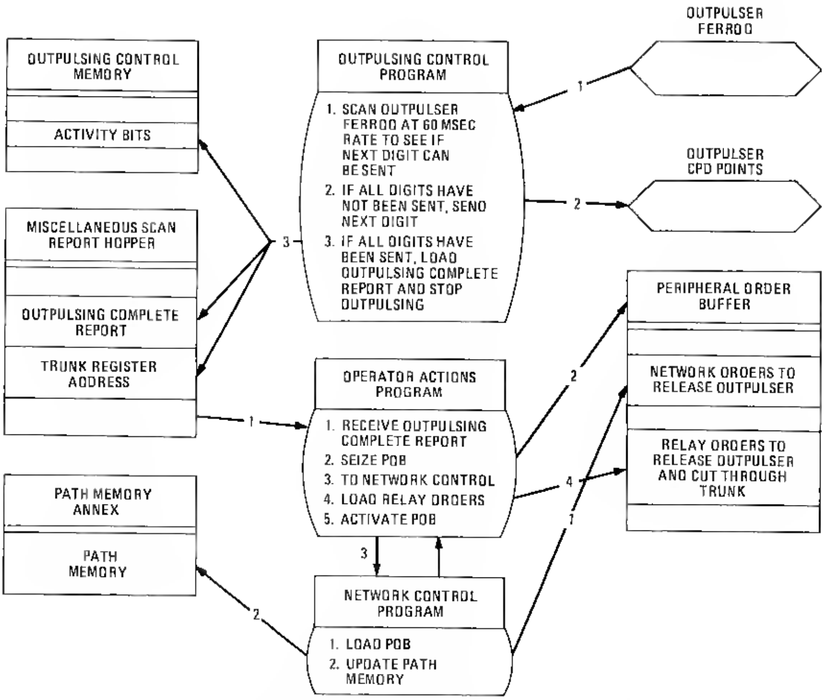


Fig. 14—Actions to release outpulser and cut through trunk.

cut-through relay of the trunk connecting the calling and called sides of the trunk. After loading these orders is completed, control is returned to the operator actions program which then activates the POB.

As shown in Fig. 15 the POB execution program causes the orders in the POB to be executed resulting in the configuration shown in Fig. 16. Upon successful completion of execution of the POB, control is returned to the operator actions program. At this time the operator actions program seizes a PIB and loads an order to extinguish the start lamp at the operator's position. This signifies to the operator that outpulsing is complete. Upon successful completion of execution of this PIB, control is returned to the operator actions program which then updates memory in the position register and trunk register to reflect that called party answer may now be expected. The temporary memory associated with the scan point of the outgoing side of the trunk is set to cause the supervisory trunk scan program to recognize when an off-hook occurs and transmit this information to the operator actions program.

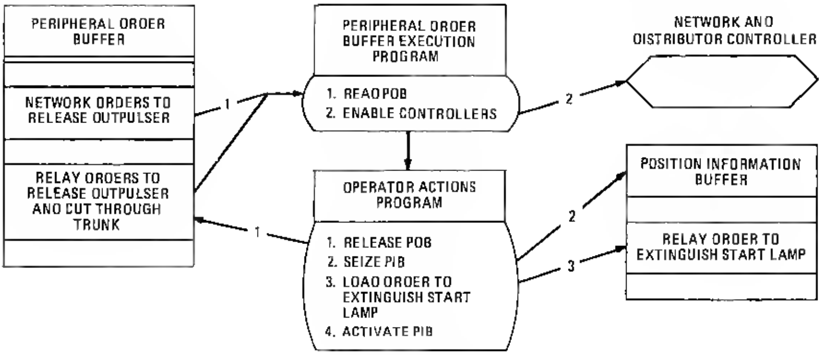


Fig. 15—Final actions to release outpulser and cut through trunk.

#### 4.6.4 *Receipt of Class of Charge*

In this example it is assumed that the next stimulus to the operator actions program is receipt of a person paid class of charge key.

Upon receipt of the person paid key, the operator actions program records in the position register that a class of charge has been received and that this particular class of charge requires called party answer before position release can be accepted. It also records in the trunk register a specific class of charge index for billing. An idle PIB is seized and an order loaded to light the person paid class of cbarge lamp to indicate to the operator receipt and acceptance of her key signal. For this example it is assumed that the operator will wait for called party answer and identification before executing any more key operations. She will also at this time request from the customer deposit of charges as displayed to her at the time her position was seized.

#### 4.6.5 *Receipt of Called Party Answer*

Upon detection of an off-hook on the outgoing side of the trunk (shown in Fig. 17), the trunk supervisory scan program loads a report in the trunk seizure and answer hopper and sets the temporary memory associated with the outgoing side ferrod to ignore any further changes of state until directed otherwise by some base level program.

Sometime later this report is unloaded from the hopper and control transferred to the operator actions program. At this time (as shown in Fig. 17) the operator actions program seizes a flash scan timing register (FSTR) and initializes it to scan the outgoing ferrod of the trunk at a 100 millisecond rate to determine if this is a true answer

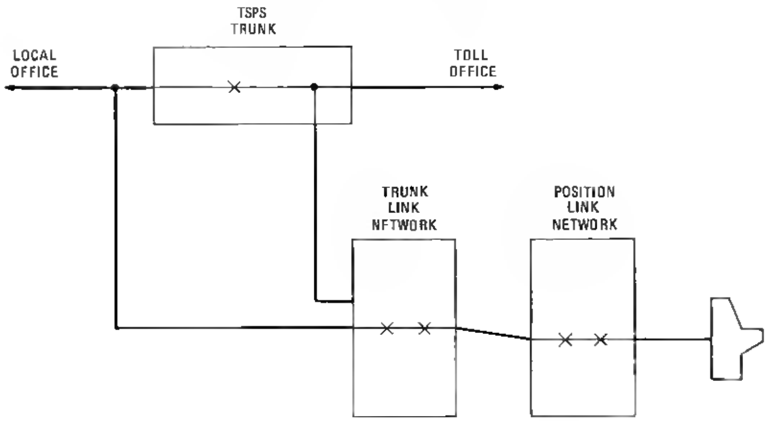


Fig. 16—Trunk cut through with position attached.

or the beginning of flashing signifying a busy or reorder condition. Upon establishing that the off-hook was in fact an answer, the flash scan timing program (as shown in Fig. 18) loads an answer report in the miscellaneous scan report hopper.

Sometime later this report is unloaded from the hopper and control returned to the operator actions program. Upon receipt of this report the operator actions program records in the position register that a called party answer has been received and executes a PIB to ex-

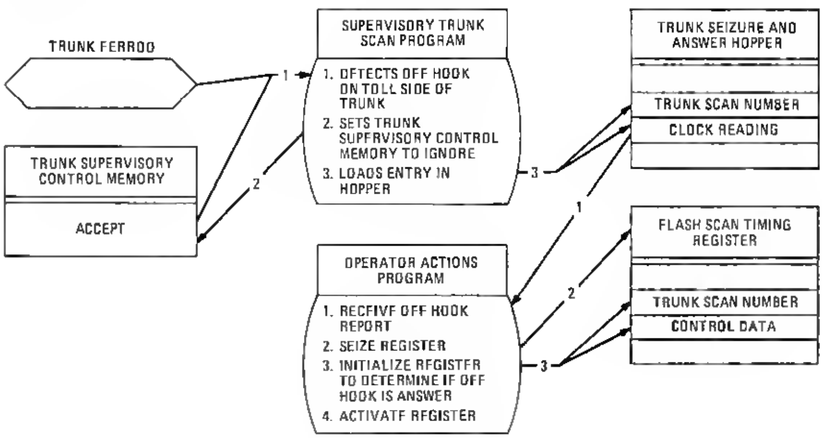


Fig. 17—Receipt of called party off hook.

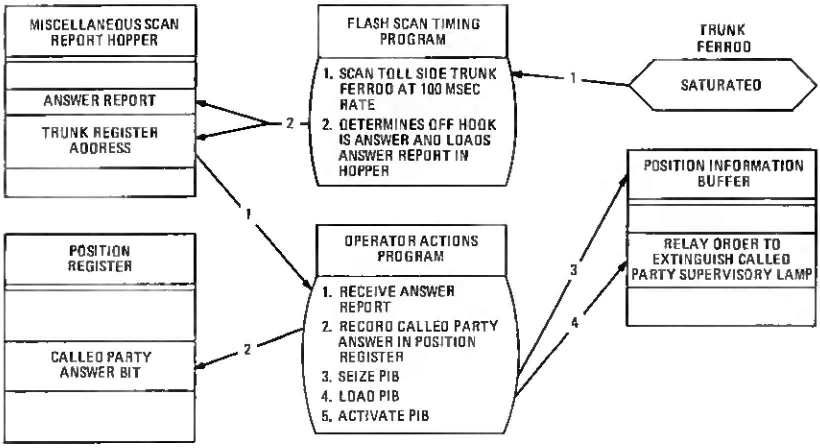


Fig. 18—Receipt of answer report.

tinguish the called supervisory lamp at the position to indicate to the operator that the called party has answered.

#### 4.6.6 *Final Operator Actions*

The operator now establishes when the desired party has been reached and conversation has started. She then depresses the start timing and position release keys.

Upon receipt of the start timing key, the operator actions program interrogates memory to establish that all necessary billing information has been provided by the operator. As this example has evolved, the result will be positive and a PIB will be executed to light the start timing lamp indicating to the operator acceptance of the key.

Upon receipt of the position release key, the operator actions program performs a sequence of tasks to release the connection of the trunk to the position. The first action (as shown in Fig. 19) is to perform a scan of the hardware clock and record this in the trunk register as the call start time. It then seizes a POB and transfers control to the network control program to load orders in the POB to release the connection to the position. After completion of the loading of these orders, control is returned to the operator actions program which loads relay orders to idle the operator cut-through circuit and then activates the POB. As shown in Fig. 20, the POB execution program causes the orders in the POB to be executed. Upon successful

execution of the POB, control is returned to the operator actions program.

At this time (as shown in Fig. 20) the operator actions program sets the trunk register memory to reflect that the trunk is now in a talking state, activates supervision on both sides of the trunk to watch for disconnect, and enters a subroutine to place the trunk register on a coin timing list. The coin timing program will later cause a return to the disconnect program approximately 18 seconds prior to the end of the initial period.

After putting the register on the timing list, the operator actions program loads a PIB and causes it to be executed which will extinguish all the lamps on the operator's position that were related to this call. Upon completion of this PIB the operator program causes the position register to be returned to the idle link list.

## V. DETECTION OF DISCONNECT

In this example it is assumed that the call terminates during initial period, prior to the initial period coin collect sequence, and is initiated by a calling party disconnect.

The supervisory trunk scan program scanning the trunk ferrods at a 100 millisecond rate detects the change of state in the incoming side

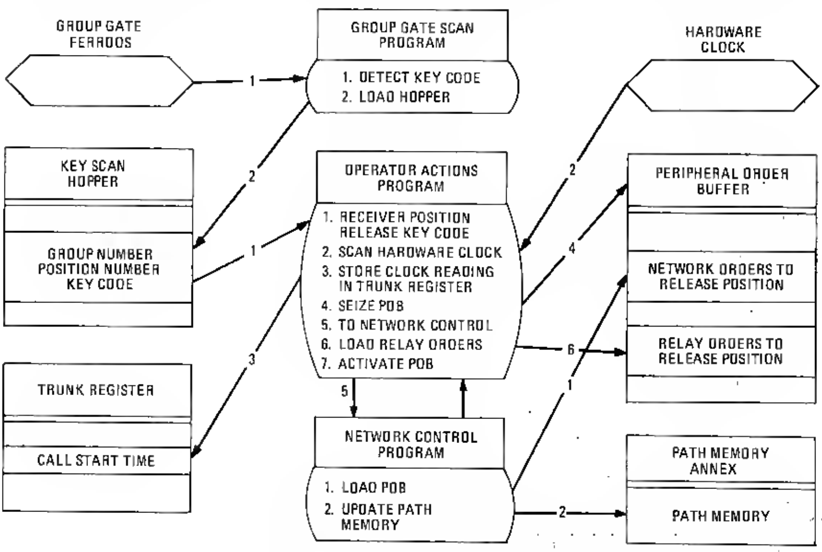


Fig. 19—Initial actions for releasing position.

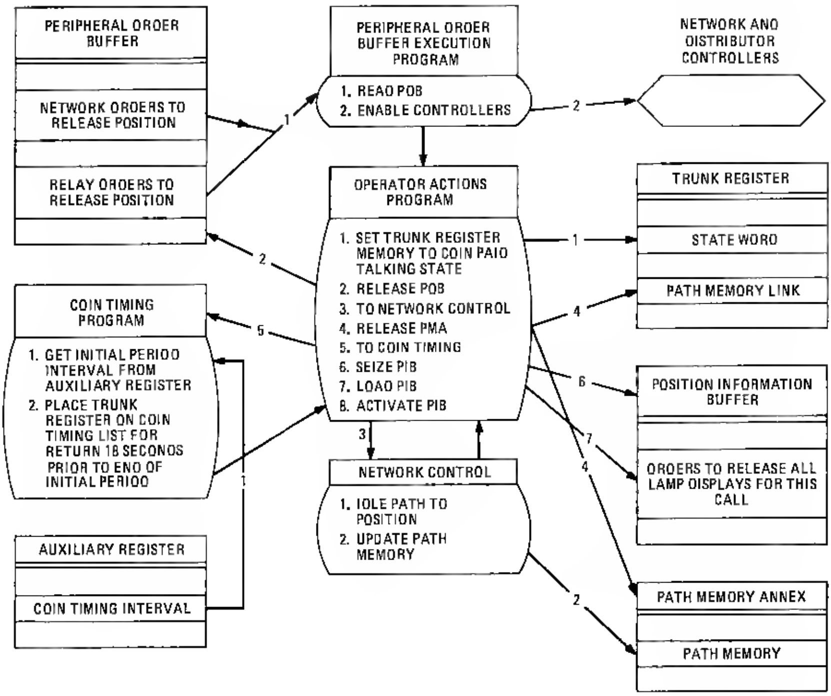


Fig. 20—Intermediate actions for releasing position.

ferrod from off-hook to tbe on-hook state as shown in Fig. 21. The supervisory trunk scan program then seizes and initializes a timed scan junior register (TSJR) by loading the trunk scan number and control data in the register to start a directed scan of the incoming ferrod.

Two hundred milliseconds later, the hit scan program reads the incoming ferrod as indicated by the trunk scan number, and if the ferrod still indicates on-hook, it recognizes a potential disconnect. The hit scan program then loads tbe trunk scan number along with the TSJR address into the hit scan result hopper with an indication that the on-hook was not a hit.

Later, the hopper entry is unloaded (as shown in Fig. 22) and control is given to the disconnect program. The disconnect program stores the disconnect time obtained from tbe TSJR in the trunk register and removes the trunk register from the coin timing list. It then begins its initial disconnect actions. An idle POB is seized and initialized, and after loading orders to release the forward connection

on the trunk, control is transferred to the network control programs to establish a connection to a coin control circuit. When these orders have been loaded, control is given to the coin trunk program, and this program completes the loading of relay orders in the POB and activates it. Upon successful execution of the POB, control is returned to the coin trunk program as shown in Fig. 23. The coin trunk program at that time seizes a multibit scan register to perform a periodic directed scan of a ferrod in the coin control circuit which indicates when the coin signal sequence of the circuit is complete. The multibit scan program performs a directed scan of the coin control circuit ferrod at a 100 millisecond rate. When the appropriate change of state is detected, the multibit scan program loads a report in the miscellaneous scan result hopper.

Sometime later (as shown in Fig. 24) the report is unloaded from the hopper and control is returned to the coin trunk program. The coin trunk program seizes an idle POB, loads orders for the idling of the coin control circuit relays, and activates the POB.

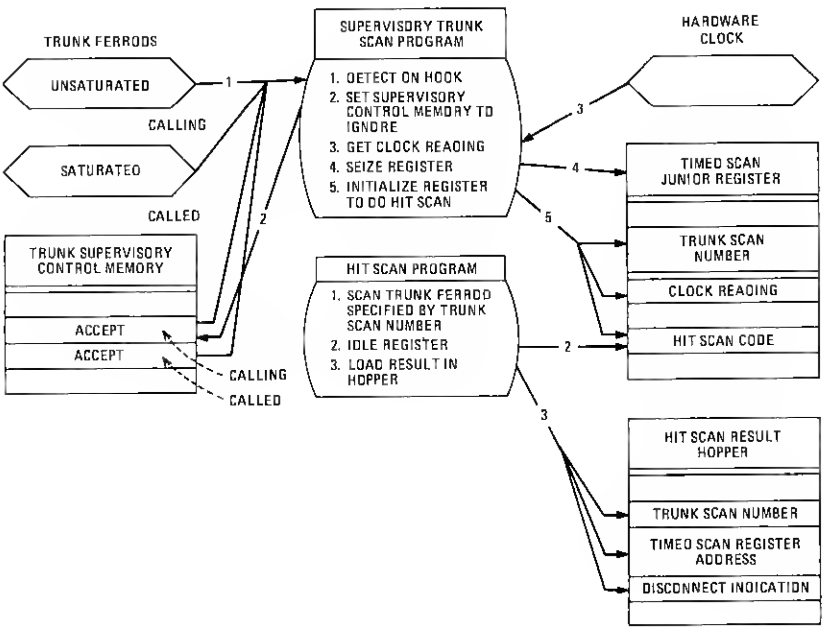


Fig. 21—Detection of on hook and hit scan.

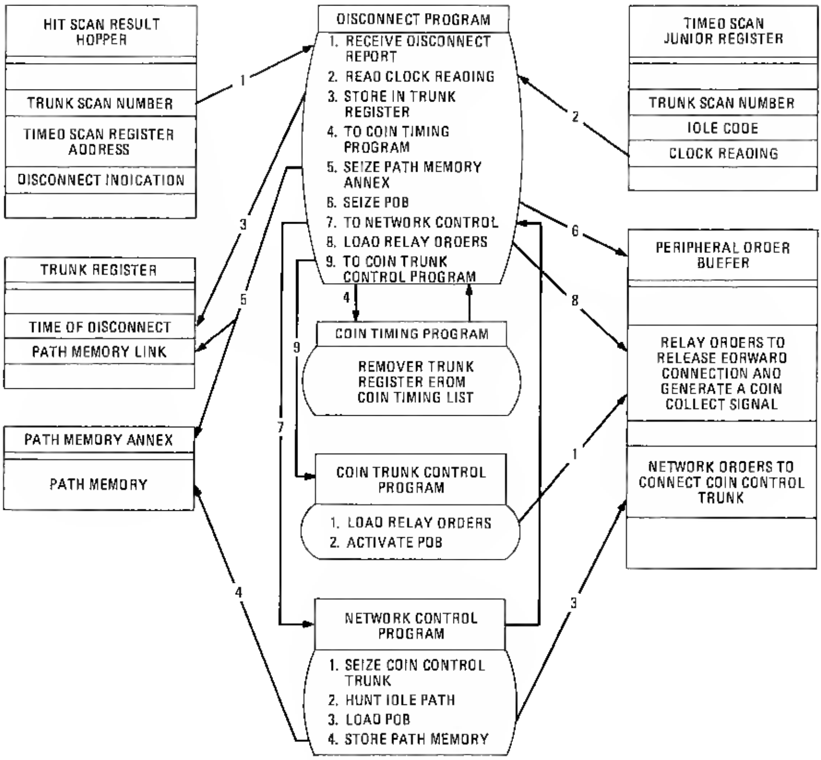


Fig. 22—Initial disconnect actions.

Upon successful completion of this POB, control is returned to the coin trunk program which then returns to the disconnect program indicating successful completion of coin actions. Using the same POB, the disconnect program transfers to the network control program which loads orders to release the connection to the coin control circuit. The disconnect program then activates the POB.

As shown in Fig. 25, after the orders in the POB are carried out, the disconnect program releases the POB and transfers to the billing accumulation program. The billing accumulation program transfers all pertinent information from the trunk register to a buffer area from which the data will be later transferred to magnetic tape. After the billing information is transferred, the disconnect program seizes an idle POB and loads orders to idle the relays of the trunk which re-

leases the off-hook condition at the outgoing trunk in the local office allowing it to return to normal and to serve new calls.

Upon successful execution of this POB, control is returned to the disconnect program as shown in Fig. 26. Since this disconnect action could have been initiated with the calling party off-hook and the local office trunk not yet returned to normal (i.e., on-hook), a scan of the incoming side ferrod is performed in order to prevent a false seizure. If the incoming ferrod is off-hook at this time, the trunk register is placed on a high and wet list where the incoming side ferrod is supervised at a 100 millisecond rate looking for an on-hook condition. When the on-hook condition is detected, idling of the trunk can be completed.

In this example, it is assumed that the calling party disconnects first. Hence, the trunk can be idled immediately. This is accomplished by initialization of the trunk register to indicate that it is in the idle state. The supervisory scan control bits are initialized to the state

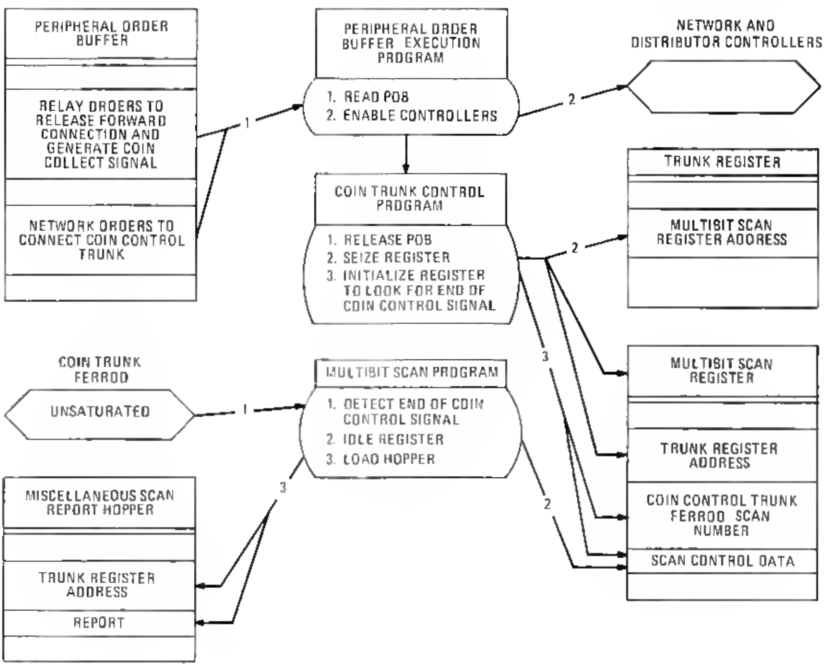


Fig. 23—Monitoring of coin control signal.


Fig. 24—Receipt of coin signal complete report.

which will permit the supervisory trunk scan program to detect a new seizure. At this time the trunk is ready to serve a new call.

## VI. AUDIT PROGRAMS

### 6.1 *Purpose of Audits*

In a program controlled system, such as TSPS, continuous system operation depends not only on the maintenance of a working hardware configuration but also on the maintenance of a working software configuration. Outages in software facilities, a condition commonly known as "data mutilation," can render the system inoperative. To guard against such mutilation, a group of programs, called audits,

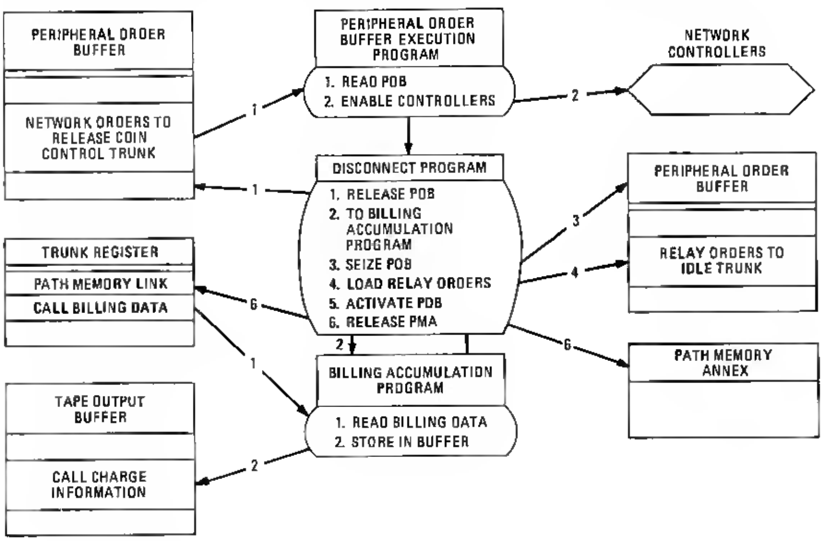


Fig. 25—Charging actions and intermediate disconnect actions.

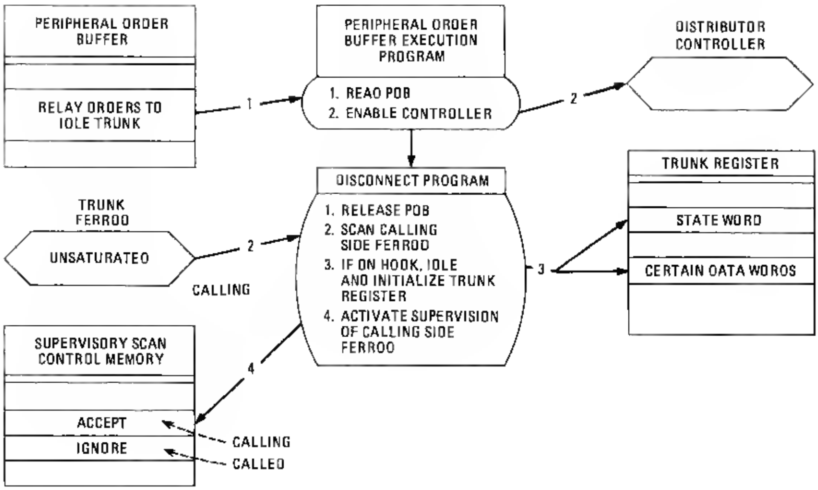


Fig. 26—Final disconnect actions.

monitors the software for error states, takes corrective action when appropriate, and initiates teletypewriter messages that record the nature and the number of the errors encountered.

Most of the audit programs run on a quasi-continuous basis as a maintenance time filler, operating whenever no other maintenance work is pending. A complete cycle of all time filler audits nominally takes one minute, with some of the more critical programs cycling on a shorter interval. Other audit programs which monitor memory areas whose error states are not critical to system operation run less frequently on an hourly or daily schedule.

#### 6.1.1 *Use in System Initialization*

Audit programs are also used for initializing memory associated with call processing and maintenance programs. In many instances, this initial state is all zeros. In other cases, such as for idle link lists of common software facilities, it is non-zero. Initialization procedures first zero all unprotected memory. The non-zero initial states are then established either by special initialization programs or by audits which recognize all zeros as an error state and consequently correct the condition by placing the memory in its initial state. Memory initialization is performed not only for the initial startup of the system but also is performed in an operating system when major outages

beyond the detection and correction capability of the audits requires special reinitialization. The latter action, known as Call Processing Recovery, is described later.

### 6.2 *Examples of Audit Programs*

The TSPS No. 1 system has about 45 distinct audit programs. The following is an example of one of these.

#### 6.2.1 *Link-List Checking*

Many call processing programs require software facilities to be structured in linked lists. For example, the software registers which are dedicated to idle multi-frequency receivers are linked as shown in Fig. 27.

The head cell, located in a permanent location, contains the memory address of the first idle register. A link word in the first register, in turn, contains the address of the second, the second of the third, and so forth. The last idle register on the list is marked by an all-zero link

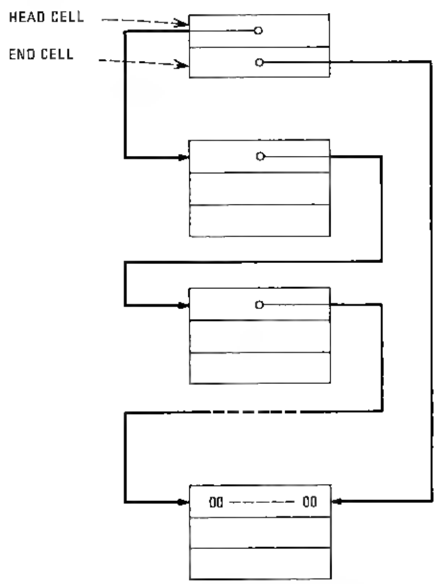


Fig. 27—Link-List Structure. Memory addresses are stored in a word of each member on the list to link that member to the next on the list. The head and end cells contain, respectively, the addresses of the first and last members on the list.

word. The end cell points to the last register. When call processing programs require an idle MF receiver, they obtain one by taking the register at the top of the list, after which the head cell is adjusted to point to the next on the list. When the programs no longer need the receiver, they restore it by placing its register at the bottom of the list, as determined by the end cell.

Audit programs maintain the list by insuring that:

(*i*) only idle registers are on the list,
(*ii*) registers not on the list are truly in use, and
(*iii*) the list structure is proper. All linkage addresses are of memory locations dedicated to multifrequency receivers, and the end cell points to the last on the list.

Any detected error state causes audits to take corrective action. The exact corrective procedure depends on the specific error detected. In general, the action is to restore all idle registers to the idle link list where they will then be available for future call needs. In the process, any register with an inconsistent state will be set to the idle state, placed on the idle link list, and the relays in its associated multifrequency receiver reset to idle. The restoral procedures do not alter any register in use on a call, provided that the software status of that register is consistent.

Since idle registers are marked by an all-zero status word, the audit which rebuilds the idle link list can also be used to initialize the idle link list. Initialization procedures will have first zeroed all scratch memory, thereby making all registers appear to be idle. Consequently, these registers will all be placed on the idle list.

### 6.2.2 *Call Processing Recovery*

On occasion, memory mutilation can become so severe that it renders the call processing system inoperative. In these instances the normal recovery action of routine audits is inadequate. Depending on the situation, they may not be able to gain control to correct the trouble. For example, when a program goes into an infinite loop, they may not be able to restore the data as fast as it is being mutilated; or they may not be powerful enough to detect the trouble. Whatever the cause, when such a situation occurs, all call processing is suspended so that the system can be devoted entirely to recovery actions. Depending on the severity of the mutilation, recovery may be performed quickly, with little disruption to service, or it may require lengthy ac-

tions with considerably more disruption, both to new calls attempting to obtain service as well as to calls being processed.

6.2.2.1 *Software Sanity Checks.* Improper software operation is detected by a set of heuristic tests, known as the Software Sanity Checks. These checks, while not capable of recognizing all possible trouble situations, are usually able to recognize improper operation. Checks are made for the following situations:

(*i*) Base-level cycling does not exceed prescribed time limits. Under normal full-load conditions entries to the base level E-priority class work are made at an average rate of once every 300 ms. If the total duration of three successive entries to this class exceeds 12 seconds, it is assumed to be due to a software problem, such as an infinite program loop.

(*ii*) Low priority J-level interrupt tasks are not called within a prescribed time limit. This check recognizes if high priority J-level activity takes an excessive amount of time, or if base-level activity is performed with the hardware in a J-level interrupt. Under these circumstances, the low priority J-level work will not be entered.

(*iii*) Relative frequency of base-level jobs is improper. A-priority jobs must be performed twice as often as B-priority jobs, B-priority jobs twice as often as C-priority jobs and so forth. If not, the system is skipping over some of its work functions.

(*iv*) Requests for interject work not answered. Base level work which must be performed within a strict time tolerance is executed upon a demand request for interject work. This check periodically makes such requests to insure that the interject mechanism is operating properly.

(*v*) Excessive maintenance interrupts occuring. This condition can be symptomatic of improper data in memory, for example, improper enable codes. If more than 15 interrupts occur within twenty seconds, this check triggers recovery operations.

(*vi*) Excessive out-of-range store addresses occuring. If memory write operations into protected (nonwritable) memory or if read or write operations of non-existent memory addresses are made, it is the direct result of data mutilation or improper program sequencing. Three such occurrences within twenty seconds triggers recovery operations.

6.2.2.2 *Description of Recovery Phases.* Recovery operations are performed in segments, known as phases. The initial request for Call

Processing Recovery brings in the lowest phase, Minor Audits. The actions of this phase are short and, as the name implies, minor. When the actions are completed, control is returned to the normal call processing programs. If the phase actions were successful, call programs will operate properly and no further Call Processing Recovery actions will be needed. But if unsuccessful, the software sanity triggers will again detect system malfunction. If the interval of call operations is less than twenty seconds, recovery proceeds to the next higher phase, Major Audits, where more extensive recovery operations are performed. Upon completion of the phase actions, normal call processing work is resumed. This process of calling in succeeding higher level phases continues until recovery is successful. The highest level phase, consequently, must be designed to be able to achieve recovery under all conceivable conditions of data mutilation.

The severity of mutilation, and hence the degree of recovery actions needed, cannot be predicted. For minor problems, recovery is rapid and disruption to call service minimal. But for major problems, recovery is more lengthy and the disruption to call service is more severe, due to the long duration of the phases and to the extent of the actions performed on calls by the phase operations. For those major problems which require the full sequence of recovery phases, overall recovery is further delayed as the system sequences through the lower level phase operations. If the severity of the problem could be predicted in advance, recovery operations for major problems could be shortened by skipping the lower level phases and jumping immediately to the highest. Unfortunately, such a prediction cannot be made.

An optimal recovery strategy, hence, must place sufficient recovery power in each phase to insure good probability of success over all possible problems. The inclusion of each phase in the recovery structure is thus based on its probability of success when compared with its effect on call service.

It is also important that the most severe system initialization phase leave as many calls undisturbed as possible without jeopardizing the effectiveness of the phase. One of the serendipitous advantages of the simple design of the TSPS universal trunk circuit permits calls that are in the customer-to-customer talking state to continue undisturbed through all phase recovery actions. The talking state is maintained by the continued operation of a single relay, and the state of the trunk can be reconstructed by determining the state of the incoming and outgoing supervisory ferrods.

The following phases have been included in the TSPS No. 1 Call Processing Recovery structure:

(i) *Minor Audits*

The lowest level phase is called Minor Audits. Its duration is of the order of a few seconds. This phase performs audits on a few short but critical software facilities. Among its actions, it verifies the enable tables, and checks all Executive Control status indicators. No call-associated data is checked in this phase. Consequently, no telephone calls are affected by its actions. However, calls in a real-time sensitive state, such as calls in a digit transmission mode, are affected by the suspension of call processing actions. These failures are not detected during the phase, but are discovered later when call processing resumes and, for the above example, recognizes that an insufficient number of digits were received. The condition is then cleared by normal call processing failure actions.

(ii) *Major Audits*

The second level of recovery phase, known as Major Audits, is nominally about thirty seconds in duration, but varies with office size. This phase performs all logical audit checks available, including the example described in Section 6.2. All calls with consistent memory states are left untouched.

For those calls which have mutilated data, the trunk ferrods are examined for off-hook states on both the calling and called sides. If both sides are off-hook, indicating a "talking connection" between customers, the connection is maintained and the software set to a state wherein the customers' eventual disconnect will be recognized. Under these circumstances, to minimize system dependence on mutilated data, all other call data associated with the call, including billing data, is destroyed. Consequently, the call continues free of charge.

(iii) *System Initialization A*

Since the Major Audit phase utilizes all available audit checks, its failure to achieve recovery signifies that the problems are uncorrectable by audits. Consequently, the next phase, System Initialization A (SIA), must initialize existing data and hence affect certain telephone calls. In this phase, all unprotected (call data) areas of memory except for a few select areas such as the system software clock are zeroed. Memory is then rebuilt to an initial state and all hardware is initial-

ized. As part of the memory rebuilding and hardware initialization process, all calls which had been in a customer-to-customer talking state are preserved, and allowed to continue free of charge. As described under the Major Audit phase, this state is recognized by noting that ferrods on both sides of the TSPS No. 1 trunk are in the off-hook state.

Although customer-to-customer talking connections are retained, other calls are disrupted. Talking connections to operators as well as all connections to digit receivers, announcement circuits, audible ringing tone, and the like are disconnected. The actions of the phase require about one minute to complete.

*(iv) System Initialization B*

Since the SIA phase places very little reliance on past data, the main cause for it to fail would be hardware troubles. Consequently, the main differences between System Initialization B (SIB) and an SIA lie in the hardware actions. In an SIB hardware initialization orders are sent over all possible, and hence redundant, paths. Since this phase is the final phase possible in the sequence, all areas of memory, including those retained on the previous phase, are zeroed. However, similar to the previous phase, calls in a customer-to-customer talking state are retained.

Because of the added hardware actions, the phase duration is longer, requiring about $1\frac{3}{4}$ minutes to complete. In accord with the assumption that the phase is needed only when hardware problems are contributing to the data mutilation, maintenance interrupts are left inhibited upon completion of the phase. This action allows the system to continue processing calls even though a peripheral hardware unit might be creating excessive interrupts.

In the unlikely event that none of these phases achieves recovery, problems other than unprotected memory mutilation would have to be corrected. Hardware outages of duplicate units, such as both processors, would naturally render the system inoperative. Should there be a mutilation of memory in the protected storage areas, which is highly unlikely, the bootstrap techniques described later under the Program Tape Unit Control programs would have to be used.

## VII. PROGRAM TAPE UNIT CONTROL

One of the major advantages of the SPC System is the ease with which the contents of memory can be changed. This feature extends over protected memory as well as unprotected storage and permits

the alteration of programs and protected data. While the teletypewriter is available to the craftsman for changes in protected data on a small scale, that instrument is unsatisfactory for many reasons when large amounts of data or program changes are necessary. The Program Tape Unit (PTU) is provided with the SPC system to facilitate large scale data and program changes as well as a tool for retrieving and storing on magnetic tape the contents of memory. This section describes the control program for the PTU and its use.

### 7.1 *On-Line Operation*

The Program Tape Unit normally operates in an on-line mode on a time-shared basis with other programs. This mode is used for both memory to tape, as well as tape to memory operations. Data transfers to and from the tape are performed every 5 ms at J-level interrupt. The transfers are made between the processor and the tape unit, with the program logic performing parity checks and software buffer loading or unloading operations. Nominally, 5 tape characters or one SPC 40-bit word are either read from or written on the tape during each interrupt.

Whereas the input/output transfer operations occur at interrupt level, the processing operations are performed at base level. If data is to be transferred to tape, a record size of 128 words, or 640 characters, is formed at base level and stored in an output buffer in memory. The tape is then started and the J-level transfer program activated. When all characters in the record have been transferred to tape, the tape is stopped and another record formed, as above. This process is repeated until all data has been transferred.

Similarly, to transfer data from tape to memory, the tape is read one record at a time by the J-level transfer program, and the characters stored in a memory buffer. After each record is transfered, the tape is stopped while the base level program checks the parity of the input, assembles the characters into 40-bit words, and updates memory or matches with memory. In the latter case, when a compare only function is being performed, all discrepancies are reported via the system maintenance teletypewriter.

Transferring data from tape to memory is performed to introduce program changes. In this mode the tape contents are written in only one memory bus system. The bus to be loaded is first forced to the standby state by controls at the Master Control Center. This manual operation results in the SPC system configuration for normal memory operations shown in Fig. 28. The loading process is performed on only

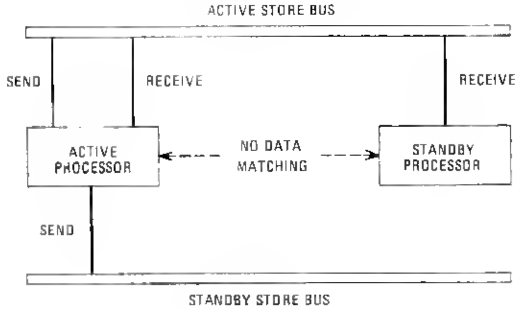


Fig. 28—SPC system configuration for memory operations.

the standby bus by use of special instructions. While the loading process is in progress, the active bus continues to process telephone calls. Since the active processor, by sending data to both store buses, maintains the unprotected memory on the standby bus, it is possible, if the changes to the standby bus are minimal, to reload memory and switch buses to the new issue without loss of call processing continuity. However, for some types of changes a system initialization (SIA) phase will have to be induced after the switch to the new bus since the scratch data in unprotected memory will probably be inconsistent with the new program changes. Normally, a new program load is made only during slack traffic hours (e.g., after midnight) to minimize the number of customers affected by the SIA. It should be noted that if the new program issue includes an expansion of protected memory, the active processor will try to write in protected memory locations on the standby bus after the protection boundary is changed. An all-seems-well write failure will result on this bus and the write will not be performed on the standby bus. But since the processors are not receiving from the standby bus, the ASW failure is ignored, and system operation is not affected.

Upon completion of loading, the standby and active buses are switched by controls from the Master Control Center, thereby activating the new program. Should the new program operate improperly, the old program can be restored by switching the unaltered bus back to the active state. The two bus systems are kept dissimilar until testing of the new program is completed to the extent made possible by the simplex bus configuration. At that time, the buses are brought into agreement by copying the contents of the new bus to the old. This is accomplished via a special store program which utilizes the normal

bus-to-bus update feature of the store diagnostic program. Then, further testing including the full store diagnostic sequence is initiated and duplex operation is restored.

Because protected memory contents are occasionally altered, primarily by the Recent Change programs, memory is copied onto tape periodically to generate a new back-up of the protected memory. This operation is typically performed monthly. The new tape thereby made is available for use should a need for reloading the memory arise and is periodically verified against the contents of memory.

### 7.2 *Off-Line Operations*

Off-line program tape unit operations are available for tape to memory transfer only. In this mode all other program activity is suspended. A special program, called the "bootstrap" program, operates in A-level interrupt to perform the sole function of loading memory from tape. Bootstrap operation is used primarily to load memory for the first time during installation of the system. In the event of a catastrophy it can be used to rewrite a mutilated program.

After a system is in operation, bootstrap is used as a last-resort recovery operation in the event that protected memory is mutilated to a degree that normal call programs cannot operate. In this event the memory back-up stored on tape is loaded to regenerate the memory. This is accomplished through use of a special bootstrap transfer card extender which is plugged into the active processor. Once inserted, a manually induced A-level interrupt at the Master Control Center causes the processor to enter the bootstrap program. In this use of the PTU, the new program or data is written into the stores on both buses simultaneously. The process is as efficient as possible to minimize the time to restore the system to normal operation. In fact, the limiting factor in the operation is the data transfer rate of the PTU itself.

When it is used to load the system for the first time during installation, the bootstrap program is manually written into memory using the test cart. The full system loading operation described above is then used to complete the operation. In order to protect against mutilation of the bootstrap program itself, which would require manually reloading the program in a working office, special precautions are taken. Four copies of the program are stored in memory with two copies on each bus. Any one of these copies can effect the loading operation in case of partial mutilation of the bootstrap program. Selection of the

copy of the bootstrap program to be used is manual and is made by means of switches on the bootstrap transfer card.

## VIII. PURPOSE OF RECENT CHANGE PROGRAMS

Within the protected area of memory are stored items known as office data parameters. These items contain data specifications which are unique to a specific office. The contents of tbese locations, while varying from one office to another, generally do not change within an office over periods of months or even years. Some locations, however, do occasionally require alterations. While the occurrence of such changes is infrequent, the advance notice is often short and the necessity great. For example, if traffic conditions change, trunks have to be added, deleted, or moved. As new telephone central offices are established and tariff changes enacted, coin rating tables must be modified.

A group of programs, known as Recent Change programs, are designed to permit the Telephone Companies to alter such office data parameters. The Recent Change programs accept functional requests for data changes, as transmitted over the Recent Change teletypewriter. The programs check the request for accuracy and alter office data tables as required.

Recent Change programs are provided for only those functions which might require variations within the normal office engineering period of two or three years. Major data updates, or recompilations, which would be required at the end of the interval will be done off-line and will not be discussed in this article.

### 8.1 *Control Structure*

Because data alterations require Recent Change programs to unlock and alter protected memory, special precautions are taken to insure that the entire change procedure is performed correctly. Alterations of protected memory locations are particularly critical. For example, an incorrect write at a program location could render the entire program system inoperative. In the event of such mutilation of protected memory, recovery could be achieved only by a "bootstrap" reload (see the section on Program Tape Unit Control).

#### 8.1.1 *Teletypewriter*

Recent Change requests are transmitted to the system over the Recent Change TTY cbannel. This teletypewriter unit is equipped

with a paper-tape reader to speed up the input operation and to increase the accuracy of tbe data tbrough off-line preparation and checking. A control cbaracter is placed on the tape at the end of each line to stop the tape reader to allow time for the Recent Change programs to process tbe input.

As each line is received, the normal TTY programs verify the format of the line, translate the input fields to a form expected by the Recent Change programs, place the result in the Recent Change Buffer, and activate the Recent Change program appropriate for that message. When the Recent Change programs have completed tbeir processing, they instruct the TTY programs to transmit back the last line (known as a "print-back"). Thus, each input line is typed twice, the first time when transmitted to the system, and the second time on the print-back. The telephone craftsman can then verify the correct transmission of the message by noting that the two lines are identical. If the Recent Change programs expect another input line to follow, they cause a control signal to be transmitted back to the teletypewriter unit to turn on the paper-tape reader. The next line is then transmitted and processed in the same manner as described above. This process continues, line by line, until all data has been received.

#### 8.1.2 *Message Sequencing*

A functional change requires multiple lines of input data. For the change to be processed correctly, the lines must be transmitted in a predetermined order. The Recent Change Control program monitors this sequence. Each change message must start with a control line, known as the "BOC," for Beginning Of Change. This line establishes the type of change expected. Only a subset of all Recent Change messages can follow the BOC. Should any otber message be transmitted, it would be rejected by tbe Recent Change Control program. Tbe programs which process each line establisb tbe set of messages whicb may legally follow.

When all lines necessary to complete a change function have been transmitted, the Recent Cbange programs are so informed by an EOC (End of Change) input message. After transmission of tbe EOC, the paper-tape reader is stopped to permit further verification. At tbis time, depending on the change, tbere may be special verify messages available to test the change. If the verification is satisfactory the entire message is then activated by typing in an "ACT" message. This message signals the system to update the permanent translation tables.

All updating is performed in duplex on both memory systems. If, however, the message is not to be activated, it can be cancelled by typing in a "CAN" message.

As the change programs process the input data, they compute the addresses of locations to be changed and the data to be stored in those locations. The addresses of the locations and the new data are stored in the Recent Change Work List. When the ACT message is received, the Recent Change Control Program uses the Work List to update permanent memory. As each word is updated, the former contents of that location, known as old data, are saved with the associated address for possible restoration.

If any maintenance problem should cause the undating process to terminate before completion, memory is restored to its prior state by restoring the old data. If any major system problems occur within fifteen minutes after completion of updating, such as a Call Processing Recovery phase or a maintenance interrupt, it is assumed that the Recent Change activity was responsible for the trouble and the change is automatically erased from the system and the old data is restored.

### 8.2 *Verification of Change Messages*

Each Recent Change message performs extensive checks on its associated input message to verify the reasonableness of the change request. Range checks are made against office parameters to insure that they are within bounds. Each input message must include information, known as old data, that describes the present status of the area to the changed. Verification of the old data insures correctness of the office records on which the change is based and insures that the processing programs are altering the correct locations.

### 8.3 *Types of Changes*

Two general types of Recent Changes can be made. New requirements at the local office can require changes to trunk circuit data, and new tariff requirements or establishment of new telephone central offices can require changes to coin rating tables. An example of the latter follows:

#### 8.3.1 *Coin Rating Data Change*

The TSPS No. 1 system provides automatic computation of charges for coin telephone calls. The calculations are made by computing air-

line distances between the originating and terminating offices and obtaining the appropriate charges for the distance, type of call (person-to-person or station-to-station) and time of day.

Extensive office data tables are stored in protected memory to provide the geographic location of all originating and terminating offices. The locations are stored as vertical and horizontal distances from a fixed reference point. Whenever new central telephone offices are placed in service, or when changes about an office in service are made, data tables have to be altered. Tbe following is the sequence of messages required to provide coin rating for a new central office:

BØC-99-ØRD:023,TYP:ADCØ,08:01:69.
RCØ-20-NXX:949,NPA:201,ØRTP:VC,NRTP:VH.
RCØ-21-DATA:NEW,VERT:05086,HØRZ:01380.
EØC-99-ØRD:023.
ACT-99-ØRD:023.
VER-48-NPA:201,NXX:949. (Not required, for vertification only.)

The BOC-99 message signifies the beginning of the change. The ORD field specifies an order number, as given by the Telephone Company. The type field, ADCO, indicates that an addition of a central office is to be made. The remaining field gives the date. The RCO-20 message identifies the new office code as 949 in the 201 numbering plan area. The program checks that the old rating type (ORTP) for the office, as stored in memory, is vacant (VC), and it changes this data to the new rating type (NRTP) of V and H (VH) for Vertical and Horizontal coordinates. The next line gives the coordinate values, as measured from a standard reference point, and identifies the data as NEW. Since the office code was previously vacant, there is no OLD coordinate data to be checked.

The EOC-99 message informs the Recent Change programs that no further data is to follow. Tbe change is activated by the ACT message, after which the change can be vertified by typing the VER-48 message. The verification program checks the office data to determine the status of the office code, and prints back a response listing the rating type and V and H vector values. The response, of course, should match the change request made in the RCO-20 and RCO-21 messages.

## IX. ACKNOWLEDGMENTS

The authors wish to acknowledge the contributions of many colleagues who took part in the development and testing of the programs

described in this article. They are too numerous to list completely here, but several deserve special mention: E. E. Hanna for his work in coin call rating and computing, Mrs. C. Millheiser, G. Giani, and S. Lederman in call processing, R. D. Kroning and H. E. Voigt in operator actions, D. J. Sidor in recent changes, and J. J. Victor in program tape unit control.